

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Белгородский государственный технологический университет
им. В.Г. Шухова

А.А. Руднев, Д.А. Юдин

**МЕТОДЫ ОБРАБОТКИ ВИДЕОИНФОРМАЦИИ В СИСТЕМАХ
ТЕХНИЧЕСКОГО ЗРЕНИЯ ПРОМЫШЛЕННЫХ РОБОТОВ С
ПРИМЕНЕНИЕМ МАТЛАВ**

Учебное пособие

Белгород
2012

УДК 004.932: 621.865.8

Авторы:
А.А. Руднев, Д.А. Юдин

Руднев А.А. Методы обработки видеоинформации в системах технического зрения промышленных роботов с применением Matlab [электронный ресурс]/ А.А. Руднев, Д.А. Юдин – Белгород: Изд-во БГТУ, 2012. – 33 с.

В учебном пособии рассмотрены различные методы обработки видеоинформации в системах технического зрения промышленных роботов с использованием программного комплекса, разработанного для системы MATLAB 5.2, включая Image Processing Toolbox.

Пособие направлено на ознакомление и изучение основных методов обработки видеоизображений в системах технического зрения (СТЗ) промышленных роботов (ПР); приобретение навыков применения этих методов посредством моделирования процесса поэтапной обработки видеоизображений.

Учебное пособие даёт представление о современных методах обработки изображений и предназначено для студентов, обучающихся по направлениям подготовки 221000.62, 220400.62, 220700.62, и по специальностям 220301.65 и 220201.65, а также может быть полезно инженерно-техническим работникам.

УДК 004.932: 621.865.8

© Белгородский государственный технологический университет (БГТУ) им. В.Г. Шухова, 2012

Содержание

1. Введение.....	4
2. Системы технического зрения низкого уровня.....	5
2.1. Получение видеоизображения.....	5
2.1.1. Квантование изображения по уровням градации серого (интенсивности).	6
2.1.2. Дискретизация изображения.	7
2.2. Предварительная обработка изображения.	7
2.2.1. Методы пространственной области.	7
2.2.2. Методы частотной области.	8
2.3. Сглаживание.....	9
2.3.1. Усреднение окрестности.	9
2.3.2. Усредненная фильтрация.	9
2.3.3. Усреднение изображения.	10
2.3.3. Сглаживающие дискретные изображения.	11
2.4. Улучшение качества изображения.....	11
2.4.1. Гистограммное выравнивание.....	11
2.4.2. Метод задания гистограммы.....	13
2.4.3. Локальное улучшение качества.....	14
2.5. Определение кромок	15
2.5.1. Метод вычисления градиента.....	16
2.5.2. Метод вычисления оператора Лапласа.....	18
3. Системы технического зрения среднего уровня.....	18
3.1. Сегментация	18
3.1.1. Проведение контуров и определение границы.	19
3.1.2. Выделение области объекта.....	20
3.2. Описание.....	22
3.2.1. Дескрипторы границы.	23
3.2.2. Дескрипторы области.	24
4. Описание функций, применяемых для реализации методов СТЗ.....	25
5. Лабораторная работа №1. Методы обработки видеоинформации в системах технического зрения промышленных роботов (с использованием программного комплекса, разработанного для системы MATLAB 5.2, включая Image Processing Toolbox)	28
5.1. Порядок выполнения работы.....	28
5.2. Пример выполнения работы	31

МЕТОДЫ ОБРАБОТКИ ВИДЕОИНФОРМАЦИИ В СИСТЕМАХ ТЕХНИЧЕСКОГО ЗРЕНИЯ ПРОМЫШЛЕННЫХ РОБОТОВ С ПРИМЕНЕНИЕМ MATLAB

1. Введение

«Зрительные» возможности робота, как и людей, обеспечиваются сложным чувствительным механизмом, который позволяет гибко и «осмысленно» реагировать на изменения внешней среды. Использование технического зрения и других методов осязания диктуется постоянной необходимостью расширять область применения робототехнических систем. Хотя датчики расстояния, тактильные датчики и датчики силы играют большую роль в совершенствовании функционирования робота, техническое зрение является наиболее мощным источником информации для робота.

Существует три вида систем технического зрения:

- 1) Измерительные. Предназначены для измерения одной или нескольких характеристик объекта по его изображению. Такие системы используются при покраске, шлифовке, работе с движущимися предметами.
- 2) Распознающие. Применяются для классификации и сортировки объектов. Нашли широкое применение в сборочных производствах.
- 3) Обзорно-информационные. Служат для организации технологического процесса посредством анализа сцен. Применяются в гибких автоматизированных производствах.

Зрение робота можно определить как процесс выделения, идентификации и преобразования информации, полученной из трехмерных изображений. Этот процесс, называемый также *техническим* или *машинным зрением*, разделяется на шесть основных этапов:

- 1) снятие информации;
- 2) предварительная обработка информации;
- 3) сегментация;
- 4) описание;
- 5) распознавание;
- 6) интерпретация.

Названные этапы удобно сгруппировать в соответствии со сложностью их реализации. Можно выделить три уровня обработки в СТЗ: низкий, средний и высокий. Хотя четких границ между этими уровнями не существует, их выделение целесообразно для классификации различных процессов, характерных для систем технического зрения.

Под низким уровнем технического зрения понимаются такие процессы, которые являются простыми с точки зрения осуществления автоматических действий, не требующих наличия искусственного интеллекта. К низкому уровню технического зрения относятся методы снятия и предварительной

обработки информации. Таким образом, этот уровень охватывает процессы, начиная непосредственно от формирования изображения и кончая процессами компенсации, такими, как уменьшение шума, а также процессами выделения простейших параметров изображения, такими, как разрывы интенсивности. Методы обработки изображений низкого уровня допускают, а зачастую и сориентированы в основном на аппаратную реализацию.

Под средним уровнем технического зрения понимаются процессы выделения, идентификации и разметки элементов изображения, полученного на нижнем уровне. С учетом вышесказанного, к системам среднего уровня относятся сегментация, описание и распознавание отдельных объектов на изображении.

Под высоким уровнем технического зрения понимаются процессы, относящиеся непосредственно к искусственному интеллекту. Методом высокого уровня можно считать интерпретацию – предположение о деталях изображения и ситуациях, неполностью идентифицируемых на прошлых этапах. Затруднения, которые могут возникнуть при обработке изображений на среднем уровне, заключаются в частичном заслонении одних предметов другими, наличии теней и бликов на изображении, проявляющихся в плохих условиях освещения.

2. Системы технического зрения низкого уровня

2.1. Получение видеоизображения

Визуальная информация преобразуется в электрические сигналы с помощью технических средств получения видеоизображений. После пространственной дискретизации и квантования по амплитуде этих сигналов получается цифровое изображение. Основными устройствами получения видеоизображений, используемыми в техническом зрении промышленных роботов, являются телевизионные камеры, в качестве чувствительных элементов которых применяются электронно-вакуумные трубки или полупроводниковые твердотельные матрицы.

Обозначим через $f(x, y)$ двумерное изображение (рис. 1), получаемое телевизионной камерой или другим устройством, дающим изображение. Здесь x и y пространственные координаты (т.е. координаты плоскости изображения), а величина f в произвольной точке (x, y) пропорциональна яркости (интенсивности) изображения в этой точке (в случае использования нецветного изображения).

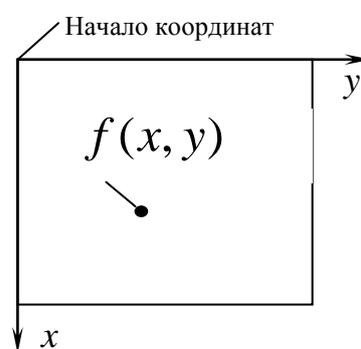


Рис. 1. Обозначения координат при описании изображения. Значение произвольной точки (x, y) задается величиной функции f в этой точке

Для получения удобной, с точки зрения вычисления, формы функции изображения $f(x, y)$ ее необходимо дискретизировать как пространственно, так и по амплитуде (интенсивности). Дискретизацию по пространственным координатам (x, y) будем называть *дискретизацией изображения*, а амплитудную дискретизацию – *квантованием по интенсивности или квантованием по уровню серого*. Последний термин применяется для одноцветных изображений и отражает тот факт, что изображение меняется по тону от черного до белого в серых оттенках.

Предположим, что непрерывное изображение дискретизировано равномерно на N рядов и M столбцов, причем каждая дискретная величина проквантована по уровню серого (интенсивности). Такая система, называемая *цифровым изображением*, может быть представлена в виде

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & f(1,1) & \dots & f(1, M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix},$$

где x и y теперь дискретные переменные: $x = 0, 1, 2, \dots, N-1$; $y = 0, 1, 2, \dots, M-1$.

Каждый элемент системы называется *элементом изображения* или *пикселем*. В соответствии с рис. 4.1 можно отметить, что $f(0,0)$ является пикселем начала координат изображения и так далее. Обычно на практике величины N, M и число уровней дискретной интенсивности каждого квантованного пиксела задают в виде целых степеней числа 2.

2.1.1. Квантование изображения по уровням градации серого (интенсивности)

Квантование изображения по уровням градации серого заключается в присвоении каждому пикселю изображения числового значения интенсивности отображаемой точки. При получении цветного изображения вместо значения интенсивности каждому пикселю присваивается кодовое значение его цвета, например, в RGB-палитре.

При получении нецветного изображения задаются массивом интенсивностей, значения которого используются для задания функции интенсивности изображения $f(x, y)$. Обычно в качестве граничных цветов выбирают белый, которому присваивают значение единицы, и черный, который кодируют нулем. Качество квантуемого изображения напрямую зависит от размерности массива, полученного путем разбиения интервала цвета $[0, 1]$ на промежуточные уровни градации серого.

При уменьшении количества уровней градации серого исходного изображения значения интенсивности каждого пикселя округляются в большую сторону к ближайшему новому значению интенсивности. При этом неизбежно снижение качества изображения за счёт частичной потери информации.

Процедура увеличения количества уровней градации серого для исходного изображения не имеет смысла. При получении черно-белого (силуэтного) изображения используются два уровня интенсивности – белый и черный. Соответственно, массив интенсивностей такого изображения может принимать два значения – нуль или единицу.

2.1.2. Дискретизация изображения.

Дискретизация изображения – это представление изображения с исходным количеством воспроизводящих точек, называемых разрешением изображения, к заданному разрешению. Например, при получении из изображения с разрешением 512×512 изображения с разрешением 256×256 необходимо уменьшить вдвое разбиение каждой пространственной координаты. Тогда размеры полученных новых пикселей будут в два раза превышать размеры пикселей исходного изображения. Интенсивность нового пикселя получается усреднением значений интенсивности окрестности размером 2×2 , где 2 – размер нового «пикселя». Уменьшение разрешения изображения – необратимый процесс.

Как правило, универсальная система технического зрения должна иметь как минимум разрешающую способность порядка 256×256 пикселей с 64 уровнями интенсивности.

2.2. Предварительная обработка изображения.

Число методов, пригодных для предварительной обработки основных параметров изображения, довольно велико, требованиям по скорости вычислений и по эксплуатационной стоимости, лежащим в основе систем технического зрения, удовлетворяет только определенная подгруппа этих методов.

Существует два основных подхода к предварительной обработке информации. Первый подход основан на методах пространственной области, а второй – на методах частотной области с использованием преобразований Фурье. Вместе эти подходы охватывают большинство из существующих алгоритмов предварительной обработки информации, применяемых в системах технического зрения роботов.

2.2.1. Методы пространственной области.

К пространственной области относится совокупность пикселей изображения. Методами пространственной области являются процедуры, оперирующие непосредственно с пикселями, составляющими рассматриваемую пространственную область. Функции предварительной обработки в пространственной области записываются в виде

$$g(x, y) = H[f(x, y)],$$

где $f(x, y)$ - входное (исходное) изображение, $g(x, y)$ - выходное (обработанное) изображение, H - оператор функции f , определенный в некоторой пространственной области, которая содержит окрестность точки (x, y) .

Основным подходом при определении окрестности точки (x, y) является использование квадратной или прямоугольной области - части изображения с центром в точке (x, y) . Центр этой области перемещается от пикселя к пикселю, начиная например, от левого верхнего угла построчно, при этом для получения $g(x, y)$ оператор применяется для каждого положения (x, y) . Квадратные формы пространственной области более предпочтительны из-за простоты их реализаций.

Один из наиболее часто встречающихся методов пространственной области основан на использовании так называемых масок свертки (или шаблонов, окон или фильтров). Обычно маска представляет собой небольшую (например, размерностью 3×3) двумерную систему, коэффициенты которой выбирают таким образом, чтобы обнаружить заданное свойство изображения.

При помощи маски, показанной на рис. 2 можно обнаружить на изображении с медленно меняющейся в пространстве интенсивностью отдельные удаленные друг от друга пиксели со значительно отличающейся от фона интенсивностью, что характерно для экстремального шума. Затем интенсивности пикселей пространственной области суммируются с заданными весовыми коэффициентами. Путём сравнения полученной суммы с некоторым пороговым значением можно определить светлее или темнее рассматриваемый пиксель области относительно фона.

	Y_{i-1}	Y_i	Y_{i+1}
X_{i-1}	-1	-1	-1
X_i	-1	+8	-1
X_{i+1}	-1	-1	-1

Рис. 2. Маска для обнаружения отдельных точек, отличающихся от постоянного фона

Данный оператор суммы, т.е. по сути, масочный оператор, можно применять для снижения шума на изображении, получения переменных порогов изображения, для формирования структуры объекта или подсчета параметров изображения.

2.2.2. Методы частотной области.

К частотной области относится совокупность комплексных пикселей в виде преобразования Фурье от изображения. Понятие частота используется при интерпретации преобразования Фурье и вытекает из того факта, что результат этого преобразования представляет собой гармонический сигнал. Из-за повышения требований к качеству обработки результатов изображения методы частотной области не так широко используются в техническом зрении роботов, как методы пространственной области.

Преобразование Фурье можно использовать при решении многих задач систем технического зрения. Одномерное преобразование Фурье является

эффективным при обнаружении движения объекта. Использование дискретного двумерного преобразования Фурье возможно при изменении, увеличении и восстановлении изображений.

2.3. Сглаживание

Операции сглаживания используются для снижения шума и других помех, которые могут появляться на изображении в результате дискретизации, квантования, передачи или возмущения внешней среды при получении изображения. Нижеследующие методы относятся к методам быстрого сглаживания и могут быть использованы в системах технического зрения промышленных роботов.

2.3.1. Усреднение окрестности.

Усреднение окрестности является прямым методом пространственной области для сглаживания изображения. Для имеющегося изображения $f(x, y)$ процесс заключается в получении сглаженного изображения $g(x, y)$, интенсивность которого в каждой точке (x, y) равна усредненному значению интенсивности пикселей функции f , содержащихся в заданной окрестности точки (x, y) . Сглаженное изображение получается при использовании соотношения:

$$g(x, y) = \frac{1}{P} \sum_{(n, m) \in S} f(n, m)$$

для всех x и y функции $f(x, y)$. Здесь S – множество координат точек в окрестности точки (x, y) , включая саму точку (x, y) ; P – общее число точек в окрестности. Степень сглаживания строго пропорциональна размерности используемой окрестности.

2.3.2. Усредненная фильтрация.

Одной из принципиальных трудностей усреднения окрестности является наличие расплывчатых изображений кромок и других характерных деталей. Эта расплывчатость часто может быть значительно уменьшена при использовании так называемых усредняющих фильтров, в которых вместо рассмотренного усреднения интенсивность каждого пикселя заменяется средним значением интенсивности в заданной окрестности пикселя.

Среднее значение M ряда величин предполагает, что часть этих величин имеет значение, меньшее M , а часть – превышающее M . Для осуществления усредненной фильтрации в окрестности пикселя сначала выделяют значения пикселя и его соседей, определяют среднее значение из существующего ряда значений интенсивностей пикселей, составляющих окрестность, и присваивают это значение рассматриваемому пикселю. Для окрестности размером 3×3 средним значением является пятая величина, для окрестности 5×5 –

тринадцатая и т.д. Когда несколько значений в окрестности равны, они группируются следующим образом. Предположим, что окрестность размерностью 3×3 содержит величины (30, 20, 20, 25, 50, 30, 20, 25, 15). Эти величины группируются в виде (15, 20, 20, 20, 25, 25, 30, 30, 50), что приводит к среднему значению 25.

Таким образом, основным назначением усредненной фильтрации является выравнивание интенсивности выделяющихся точек с интенсивностью соседних точек. Это позволяет устранить появляющиеся в поле маски фильтра изолированные пики интенсивности.

2.3.3. Усреднение изображения.

Для математического описания метода усреднения изображения рассмотрим изображение с помехами $g(x, y)$, которое состоит из суммы шумового сигнала $n(x, y)$ и неискаженного изображения $f(x, y)$, т.е.

$$g(x, y) = f(x, y) + n(x, y),$$

где предполагается, что шум не коррелирован и имеет нулевое среднее значение. Целью следующей процедуры является получение сглаженного изображения путем задания ряда изображений с помехами $g_i(x, y)$, $i = 1, 2, \dots, K$.

Если характеристики шума удовлетворяют приведенным ограничениям, то с помощью статистических методов несложно показать, что при формировании изображения $\bar{g}(x, y)$ усреднением K различных изображений с помехами

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y)$$

получим

$$E\{\bar{g}(x, y)\} = f(x, y)$$

и

$$\sigma_{\bar{g}}^2(x, y) = \frac{1}{K} \sigma_n^2(x, y),$$

где $E\{\bar{g}(x, y)\}$ - ожидаемое значение \bar{g} , близкое к неискаженному изображению; $\sigma_{\bar{g}}^2(x, y)$ и $\sigma_n^2(x, y)$ отклонения \bar{g} и n . Стандартное отклонение любой точки в усредненном изображении равно

$$\sigma_{\bar{g}}(x, y) = \frac{1}{\sqrt{K}} \sigma_n(x, y).$$

Поскольку $E\{\bar{g}(x, y)\} = f(x, y)$, то $\bar{g}(x, y)$ приближается к неискаженному изображению $f(x, y)$ при увеличении числа изображений с помехами, используемых в процессе усреднения.

К недостаткам данного метода относится то, что все объекты в рабочем пространстве должны быть неподвижны относительно камеры в процессе усреднения. Многие системы технического зрения имеют возможность выполнения полного цикла сложения изображений за один дискретный интервал времени. Таким образом, усреднение нескольких изображений займет некоторое время, в течение которого не должно быть движения объектов.

2.3.3. Сглаживающие дискретные изображения

Дискретные изображения получаются при использовании контурного или структурированного освещения или при определении пороговых значений. При этом темные пиксели обозначаются через 1, а светлые – через 0. Определение темного или светлого пиксела происходит посредством сравнения интенсивности пиксела и некоторого заданного порогового уровня интенсивности. Поскольку после такого преобразования изображение состоит из двух уровней градации серого, помехи могут привести к искажениям границ объектов, стёртых углов, появлению на объекте или на фоне участков с отличной интенсивностью, что может привести к неверному определению формы и размеров объекта и выделению ложных объектов на изображении.

Основная идея метода состоит в определении булевой функции, вычисляемой в окрестности с центром в пикселе p , и в присвоении пикселю p значения 1 или 0 в зависимости от пространственного расположения и дискретных значений соседних пикселов. Для устранения последствий проявления перечисленных дефектов на дискретном изображении применяются некоторые булевы функции, либо алгоритмические последовательности таких функций.

2.4. Улучшение качества изображения

Одной из принципиальных трудностей во многих задачах систем технического зрения нижнего уровня является необходимость автоматической адаптации при изменении освещения. Возможность компенсации таких эффектов, как тени и блики на изображении, часто является решающей для успешного выполнения дальнейшего алгоритма обработки информации. Ограничимся дискретными методами, пригодными для систем технического зрения промышленных роботов. Под пригодностью здесь понимается наличие высокого быстродействия и средней сложности используемого оборудования.

2.4.1. Гистограммное выравнивание.

Пусть переменная r обозначает интенсивность пикселей корректируемого изображения. Сначала предположим, что r – нормализованная непрерывная переменная, находящаяся в диапазоне $0 \leq r \leq 1$. Для любого r на интервале $[0,1]$ выполняется преобразование вида:

$$s = T(r),$$

которое определяет результирующее значение интенсивности s для прежнего значения интенсивности r каждого пикселя в исходном изображении. Это предполагает, что функция преобразования T удовлетворяет условиям:

1. $T(r)$ – однозначная и монотонно возрастающая функция на интервале $0 \leq T(r) \leq 1$.
2. $0 \leq T(r) \leq 1$ для $0 \leq r \leq 1$.

Первое условие исключает переход с черной шкалы интенсивности на белую, а второе условие гарантирует нахождение интенсивности изображения в диапазоне от 0 до 1 для всех значений пикселей. Изображённая на рис. 3 функция в случае применения к исходному изображению осуществляет его «компрессию», т. е. смещает серое изображение, к которому чаще всего относится интересующий объект, в область наибольшего разрешения по уровню интенсивностей.

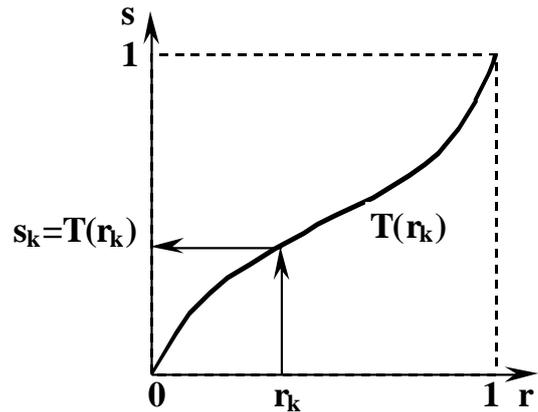


Рис. 3. Функция преобразования интенсивности

Интенсивности r и s распределены на интервале $[0, 1]$ случайным образом и, следовательно, могут характеризоваться с помощью соответствующих функций плотности распределения вероятности $p_r(r)$ и $p_s(s)$. Именно от характера поведения соответствующей функции плотности распределения вероятности на интервале $[0, 1]$ зависит основная тональность изображения.

Если большинство пикселей изображения имеют интенсивность близкую к нулю, то имеет место изображение с темной тональностью. Если интенсивности большего количества пикселей имеют значения, близкие к единице, то тональность всего изображения светлая. Функции плотности распределения вероятности для описанных случаев приведены на рис. 4.

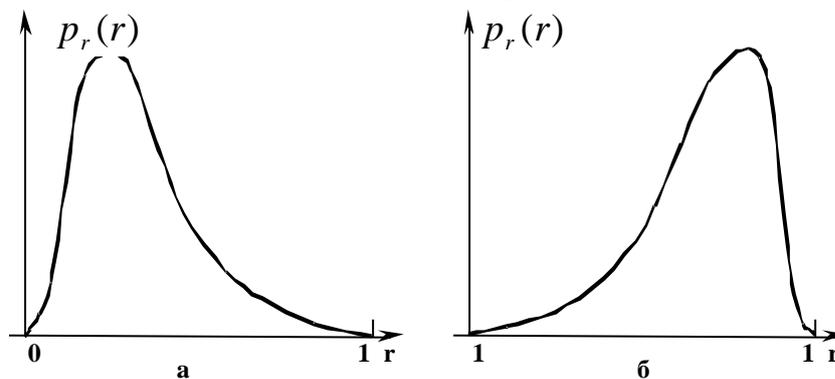


Рис. 4. Функции плотности распределения вероятности темного (а) и светлого (б) изображений

Основной эффект рассмотренного преобразования заключается в выравнивании распределения интенсивностей пикселей изображения, полученного системой технического зрения промышленного робота. Это существенно влияет на качество изображения и позволяет избавиться от таких дефектов изображения, как постоянное затемнение или, наоборот, засвечивание изображения. Данное преобразование идеально соответствует автоматической коррекции.

Для использования в цифровой обработке применяются следующие соотношения:

$$p_r(r_k) = \frac{n_k}{n}, \quad 0 \leq r_k \leq 1, \quad k = 0, 1, 2, \dots, L-1,$$

где L – число дискретных уровней интенсивности; $p_r(r_k)$ – оценка вероятности интенсивности r_k , трактуемая как частота проявления на изображении пикселей с интенсивностью r_k ; n_k – число появлений данной интенсивности на изображении, а n – общее число пикселей на изображении. График зависимости $p_r(r_k)$ называют гистограммой, а метод, используемый для получения равномерной гистограммы, известен как гистограммная линейаризация (рис. 5).

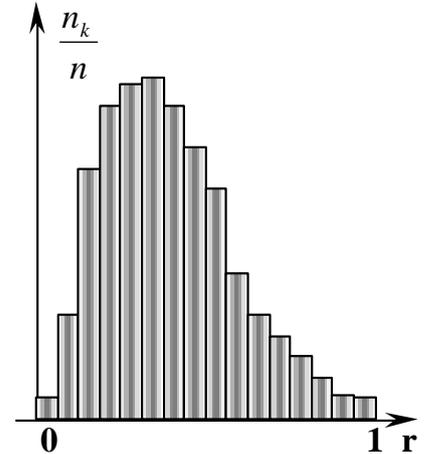


Рис. 5. Пример гистограммы

Гистограмма в таком случае будет представлять собой график, состоящий из набора вертикальных столбцов, расположение которых по оси абсцисс соответствует значениям интенсивностей пикселей изображения, а высота столбцов показывает отношение количества пикселей с данной интенсивностью к общему количеству пикселей исходного изображения.

Простейшая дискретная форма реализации функции $T(r)$ для улучшения качества затемнённых изображений имеет вид:

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j),$$

а для засвеченных изображений:

$$s_k = T(r_k) = \sum_{j=k}^1 \frac{n_j}{n} = \sum_{j=k}^1 p_r(r_j).$$

Эти зависимости показывают, что для получения значения s_k , соответствующего r_k , производится простое суммирование элементов гистограммы от 0 до r_k , либо r_k до 1.

2.4.2. Метод задания гистограммы

Гистограммное выравнивание удобно для автоматической коррекции, поскольку оно основано на функции преобразования, т.е. однозначно определяется с помощью гистограммы исходного изображения. Однако этот метод ограничен в том смысле, что в нем используется только функция гистограммной линейаризации. Этого оказывается недостаточно, когда исходной информацией является желаемая форма выходной гистограммы.

На практике метод наиболее широко применяется в случаях, когда полученное изображение имеет сразу несколько дефектов, например, при съем-

ке издали освещенной сцены полученное изображение может содержать засвеченную от источника освещения область и глубокую тень от предметов. Чтобы избавиться от таких недостатков изображения, необходимо к исходному изображению применить метод задания гистограммы. Характер заданной гистограммы в таком случае будет выглядеть как показано на рис.6. Гистограмма такой формы позволяет линеаризовать, или уравнять, соотношение темных и светлых пикселей на изображении.

Алгоритм метода можно обобщить следующим образом:

1. Выровнять уровни интенсивности исходного изображения с помощью уравнения

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j).$$

2. Определить желаемую ФПВ интенсивности и получить функцию преобразования

$$G(z_i) = \sum_{j=0}^i p_z(z_j).$$

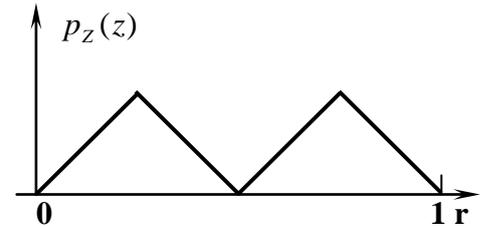


Рис.6. Форма задания гистограммы.

3. Осуществить обратное преобразование $z_i = G^{-1}(s_i)$ к уровням интенсивности изображения с выровненной по п.1 гистограммой.

В предложенных уравнениях $p_r(r_j)$ вычисляется из исходного изображения, а $p_z(z_j)$ – задается. Смысл данных уравнений следующий. Выравниваются уровни интенсивности заданного изображения по первой формуле и задаются желаемые оценки вероятностей интенсивности $p_z(z_j)$ получаемого изображения. Затем необходимо получить функцию преобразования $G(z)$, используя второе уравнение. После этого в исходном изображении осуществить замену пикселей с интенсивностью r_j на интенсивность z_i .

2.4.3. Локальное улучшение качества

Приведенные методы гистограммного выравнивания и задания гистограммы являются общими в смысле изменения интенсивности пикселей с помощью функции преобразования на всем поле изображения. Хотя эти методы дают общее улучшение качества, часто необходимо выделить детали на достаточно малых участках изображения. Поскольку качество пикселей на этих участках может мало влиять на вычисления при преобразовании всего изображения, использование общих методов редко приводит к улучшению качества на малых участках. Решением данной проблемы является усовершенствование функций преобразования, основанных на распределении интенсивности или на других свойствах, в окрестности каждого пикселя исходного изображения.

Рассмотренные выше гистограммные методы легко трансформируются для локального улучшения качества. Трансформация заключается в определении окрестности размерностью $n \times m$ и в перемещении ее центра от пикселя

к пикселю. В каждом положении вычисляется гистограмма точек в данной окрестности и определяется функция преобразования гистограммного выравнивания или функция преобразования метода задания гистограммы. Эта функция служит для отображения интенсивности пикселя, расположенного в центре окрестности. Затем центр участка размерностью $n \times m$ перемещается на соседний пиксель и процесс повторяется. Такой алгоритм вычислений интенсивности каждого пикселя области позволяет вести расчеты с новыми параметрами, появляющимися на каждом шаге движения.

Качество изображения на локальных участках можно улучшить, используя и другие параметры интенсивностей пикселей, например, среднее значение интенсивности и изменение интенсивности (или стандартное отклонение). Среднее значение – это мера средней яркости пикселей в окрестности, а изменение интенсивности – мера контрастности изображения. Типичное локальное преобразование, основанное на этих параметрах, переводит интенсивность исходного изображения $f(x, y)$ в интенсивность нового изображения $g(x, y)$ путем осуществления следующей операции над расположением (x, y) каждого пикселя:

$$g(x, y) = A(x, y)[f(x, y) - m(x, y)] + m(x, y),$$

$$\text{где } A(x, y) = k \frac{M}{\sigma(x, y)} \text{ для } 0 < k < 1.$$

В этой записи $m(x, y)$ и $\sigma(x, y)$ – соответственно среднее значение интенсивности и стандартное отклонение, подсчитанные для окрестности с центром в точке (x, y) ; M – среднее значение всей функции $f(x, y)$, а k – постоянная, выбираемая из указанного диапазона.

Важно отметить, что A , m и σ – переменные величины, которые зависят от заданной окрестности точки (x, y) . Локальные изменения увеличиваются за счет умножения разности между $f(x, y)$ и локальным средним значением на локальный коэффициент $A(x, y)$. Поскольку величина $A(x, y)$ обратно пропорциональна стандартному отклонению интенсивности, участки с низкой контрастностью имеют большее усиление. На практике желательно подставлять часть среднего значения и ограничивать изменение функции $A(x, y)$ в пределах $[A_{MIN}, A_{MAX}]$ для уравнивания больших отклонений интенсивности на отдельных участках. В данной лабораторной работе методы локального улучшения качества не реализованы в силу их громоздкости и сложности пересчета.

2.5. Определение кромок

Определение кромок занимает центральное место при обработке информации в системах технического зрения, представляя собой начальный этап реализации многочисленных алгоритмов идентификации объектов.

Основным принципом большинства методов является вычисление частных производных. На рис.7 приведены изображение светлого объекта простой формы на темном фоне, кривая интенсивности вдоль горизонтальной линии сканирования изображения, а также первая и вторая производные этой кривой (то же и для темного объекта на светлом фоне). Отметим, что участки кривой, соответствующие кромке, представляют собой наклонные, а не вертикальные линии, какие должны быть в случае резкого изменения интенсивности. Это отражает тот факт, что кромки на цифровом изображении в результате дискретизации обычно слегка размыты. В указанной модели кромки первая производная всех участков кривой с постоянной интенсивностью равна нулю и является постоянной величиной на участках изменения интенсивности. С другой стороны, вторая производная равна нулю на всех участках, кроме начальных и конечных точек изменения интенсивности. С учетом этого очевидно, что величина первой производной может быть использована для обнаружения наличия кромки, а знак второй производной – для определения того, какой осуществляется переход – со светлого участка на темный, или наоборот.

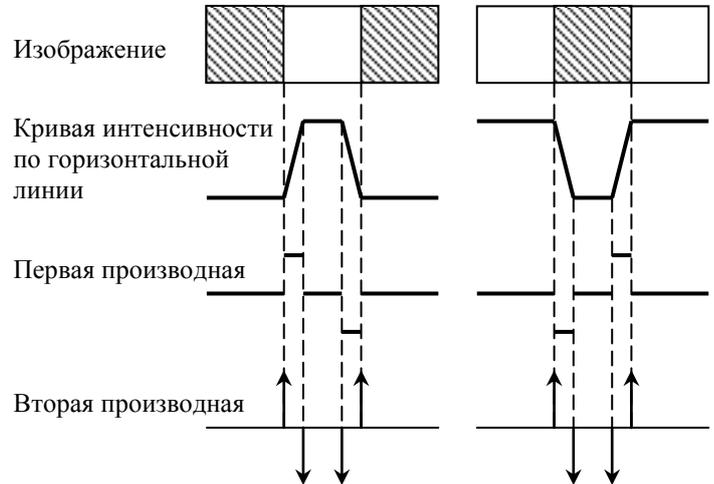


Рис. 7. Определение кромок с помощью производных.

2.5.1. Метод вычисления градиента.

Градиент изображения $f(x, y)$ в точке (x, y) определяется как двумерный вектор:

$$G(f(x, y)) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{d}{dx} f(x, y) \\ \frac{d}{dy} f(x, y) \end{bmatrix}.$$

Из векторного анализа известно, что вектор G указывает направление максимального изменения функции f в точке (x, y) . Однако, при определении кромок наибольший интерес представляет величина этого вектора, называемого обычно градиентом и обозначаемого как $G[f(x, y)]$, где

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2} = \left[\left(\frac{\partial f(x, y)}{\partial x} \right)^2 + \left(\frac{\partial f(x, y)}{\partial y} \right)^2 \right]^{1/2}.$$

На практике, как правило, градиент аппроксимируется абсолютными значениями

$$G[f(x, y)] \cong |G_x| + |G_y|.$$

Эта аппроксимация значительно упрощает реализацию метода, в особенности при использовании распространенного оборудования.

Вычисление градиента основано на нахождении первых производных функции изображения по каждой из пространственных координат. Для цифрового изображения это можно сделать несколькими путями. Несколько более сложный способ включает пиксели в окрестности размерностью 3×3 с центром в точке (x, y) . Вычисление градиента в такой области дает большее усреднение, что делает градиент менее чувствительным к помехам. Причем, использование размерности 3×3 обеспечивает большую скорость вычисления.

$$G_x = \frac{d}{dx} f(x, y) = [f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)]$$

$$G_y = \frac{d}{dy} f(x, y) = [f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)]$$

В соответствии с изложенным, значение G_x можно определить с помощью маски, представленной на рис. 8, а, значение G_y , соответственно, по маске на рис. 8, б. Эти две маски обычно называются *операторами Собеля*. Перемещая маски по изображению $f(x, y)$, получают градиенты во всех его точках.

	y_{i-1}	y_i	y_{i+1}
x_{i-1}	-1	-2	-1
x_i	0	0	0
x_{i+1}	1	2	1

а

	y_{i-1}	y_i	y_{i+1}
x_{i-1}	-1	0	1
x_i	-2	0	2
x_{i+1}	-1	0	1

б

Рис. 8. Маски для вычисления G_x (а) и G_y (б).

После вычисления градиентов по каждой из пространственных координат получить исходное изображение можно путем объединения полученных изображений каждого из градиентов.

Существует множество способов формирования выходного изображения $g(x, y)$, основанных на вычислении градиента. Простейшим способом является задание функции g в точке (x, y) значения, равного величине градиента входного изображения f в этой точке, т.е.

$$g(x, y) = G[f(x, y)].$$

Другим способом получения дискретного изображения является применение следующих соотношений:

$$g(x, y) = \begin{cases} 1 & \text{при } G(f(x, y)) > T \\ 0 & \text{при } G(f(x, y)) < T \end{cases}$$

где T – неотрицательная пороговая величина. В этом случае имеют значение только те пиксели кромки, градиенты которых превышают величину T . Таким образом, использование приведенного соотношения может рассматриваться как процесс выделения только тех пикселей, которые характеризуются значительным перепадом интенсивности.

В данной лабораторной работе реализовано вычисление градиентов изображения отдельно по каждой координате с задаваемым порогом срабатывания градиента и дальнейшее объединение этих двух изображений с целью получения границы предмета.

$$g(x, y) = G_x(x, y) \vee G_y(x, y).$$

Подбирая значения порога срабатывания градиента, можно подобрать такое его значение, при котором граница изображения получается ровной, без разрывов.

2.5.2. Метод вычисления оператора Лапласа.

Оператор Лапласа является оператором производных второго порядка вида

$$L[f(x, y)] = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}.$$

Для дискретного изображения применяют оператор Лапласа, определяемый в окрестности размерностью 3×3 с центром в точке (x, y) следующим образом

$$L[f(x, y)] = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y).$$

Эта цифровая запись оператора Лапласа дает нуль на участках постоянной интенсивности и на участках изменяющейся интенсивности на кромках, что характерно для производной второго порядка.

Однако, оператор Лапласа, хотя и определяет переход интенсивности, редко сам по себе используется для нахождения кромки, поскольку дает результат обычно очень чувствительный к помехам. Таким образом, этот оператор, как правило, выполняет второстепенную роль при определении, на какой из сторон (темной или светлой) кромки находится данный пиксель. В данной лабораторной работе метод определения кромок при помощи оператора Лапласа не рассматривается.

3. Системы технического зрения среднего уровня

3.1. Сегментация

Сегментацией называется процесс подразделения сцены на составляющие части или объекты. Сегментация является одним из основных элементов работы автоматизированной системы технического зрения, так как именно на этой стадии обработки объекты выделяются из сцены для дальнейшего распознавания и анализа. Алгоритмы сегментации, как правило, основываются

на двух фундаментальных принципах: разрывности и подобии. В первом случае основной подход основывается на определении контуров, а во втором – на определении порогового уровня и расширении области. Эти понятия применимы как к статическим, так и к динамическим сценам.

Таким образом, формально процесс сегментации – это процесс выделения на изображении интересующих объектов. Выделение объектов может быть реализовано многими способами в зависимости от того, с какой целью это делается. Так, например, для распознавания объекта из ряда предложенных можно воспользоваться только контуром данного объекта, а можно провести сравнение по нескольким пунктам, для чего необходимо иметь полное изображение объекта.

3.1.1. Проведение контуров и определение границы.

В идеальном случае рассмотренные ранее методы СТЗ низкого уровня определяют пиксели, лежащие на границе между объектом и фоном. На практике данный ряд пикселей редко полностью характеризует границу из-за шума, разрывов на границе вследствие неравномерной освещенности и других эффектов, приводящих к размытию изображения. Таким образом, алгоритмы обнаружения контуров сопровождаются процедурами построения границ объектов из соответствующих последовательностей пикселей. Различают локальный и глобальный анализ изображения на предмет построения непрерывной границы изображения.

Локальный анализ. Одним из наиболее простых подходов соединения точек контура является анализ характеристик пикселей небольшой в небольшой окрестности (например, в окрестности размером 3×3 или 5×5) каждой точки (x, y) образа, который уже подвергся процедуре обнаружения контура. Все точки, являющиеся подобными (критерий подобия определяется ниже), соединяются, образуя границу из пикселей, обладающих некоторыми общими свойствами, позволившими выделить необходимый критерий подобия.

При таком анализе для установления подобия пикселей контура необходимо определить: 1) величину градиента, требуемого для построения контурного пикселя, и 2) направление градиента. Первая характеристика обозначается величиной $G[f(x, y)]$ и определяется в п. 2.5.1. Таким образом, пиксель контура с координатами (x', y') подобен по величине в определенной ранее окрестности (x, y) пикселю с координатами (x, y) , если справедливо неравенство

$$|G[f(x, y)] - G[f(x', y')]| \leq T,$$

где T – пороговое значение.

Направление градиента устанавливается по углу вектора градиента, т.е.

$$\theta = \arctg \left[\frac{G_y}{G_x} \right],$$

где θ – угол (относительно оси x), вдоль которого скорость изменения имеет наибольшее значение. Тогда можно сказать, что угол пикселя контура

с координатами (x', y') в некоторой окрестности (x, y) подобен углу пикселя с координатами (x, y) при выполнении следующего неравенства:

$$|\theta - \theta'| < A,$$

где A – пороговое значение угла. Необходимо отметить, что направление контура в точке (x, y) в действительности перпендикулярно направлению вектора градиента в этой точке. Однако, для сравнения направлений это неравенство дает эквивалентные результаты.

Основываясь на этих предположениях, соединяются точка в некоторой окрестности (x, y) с пикселем, имеющим координаты (x, y) , если удовлетворяются критерии по величине и направлению. Двигаясь от пикселя к пикселю и представляя каждую присоединяемую точку как центр окрестности, процесс повторяется для каждой точки образа. Интенсивность вновь присоединенных к изображению пикселей выбирается таким образом, чтобы они визуально продолжали линии контура и отличались от цвета фона изображения.

Глобальный анализ. Изложенный выше метод основан на задании последовательности точек контура, полученных в результате градиентного преобразования. Этот метод редко применяется для предварительной обработки данных в ситуациях, характеризующихся высоким уровнем шума, вследствие того, что градиент является производной и усиливает колебания интенсивности.

Существует глобальный подход, основанный на представлении сегментов контура в виде графа и в поиске на графе пути наименьшей стоимости, который соответствует значимым контурам. Этот подход представляет собой приближенный метод, эффективный при наличии шума. Процедура применения метода графов значительно сложнее и требует значительно больше времени обработки, чем метод, изложенный выше. Задача поиска пути наименьшей стоимости по графу может быть решена при помощи динамического программирования.

3.1.2. Выделение области объекта.

Целью сегментации является разделение объекта на области. Если в предыдущем пункте данная задача интерпретировалась и решалась как нахождение границ между областями, то в этом разделе следует рассмотреть методы сегментации, основанные на прямом нахождении областей.

Расширение области за счет объединения пикселей. Расширение области сводится к процедуре группирования пикселей или подобластей в большие объединения. Простейшей из них является процедура *агрегирования пикселей*. Процесс начинается с выбора множества узловых точек, с которых происходит расширение области в результате присоединения к узловым точкам соседних пикселей с похожими характеристиками (интенсивность, текстура или цвет). Выбор основывается на априорной информации.

Свойство P , которое используется для того, чтобы отнести пиксель к той или иной области, состоит в том, что модуль разности между интенсивностями пикселя и узловой точки не превышает пороговый уровень T . Любой пиксель, удовлетворяющий этому свойству одновременно для обоих узлов, произвольно попадает в область R_1 .

На рис. 9 показан пример расширения области на основе выбора узловых точек. Цифры внутри ячеек указывают интенсивность. Пусть точки с координатами $(3, 2)$ и $(3, 4)$ используются как узловые (см. рис.9, а). Выбор двух начальных точек приведет к сегментации образа на две области: области R_1 , связанной с узлом $(3, 2)$, и области R_2 , связанной с узлом $(3, 4)$. На рис. 9,б показан результат, полученный для порогового уровня $T = 3$. Точки области R_1 показаны символами «+», а точки области R_2 – символами «*». Если же, с другой стороны, выбрать $T = 8$, то в результате применения алгоритма сегментации будет получена единственная область (см. рис. 9, в).

	1	2	3	4	5
1	0	0	5	6	7
2	1	1	5	8	7
3	0	<u>1</u>	6	<u>7</u>	7
4	2	0	7	6	6
5	0	1	5	6	5

а

	1	2	3	4	5
1	+	+	*	*	*
2	+	+	*	*	*
3	+	+	*	*	*
4	+	+	*	*	*
5	+	+	*	*	*

б

	1	2	3	4	5
1	+	+	+	+	+
2	+	+	+	+	+
3	+	+	+	+	+
4	+	+	+	+	+
5	+	+	+	+	+

в

Рис. 9. Пример расширения области на основе использования известных начальных (узловых) точек.

Представленный пример, несмотря на его простоту, иллюстрирует некоторые важные проблемы расширения области. Двумя очевидными из них являются: выбор начальных узлов для правильного представления областей и определение подходящих свойств для включения точек в различные области в процессе расширения. Массив узловых точек выбирается на основе априорной информации, которая и определяет характер критерия объединения пикселей. Например, если имеется снимок температурного поля объекта, то за узлы можно взять точки с максимально отличной температурой от температуры фона. При отсутствии априорной информации можно начать с вычисления для каждого пикселя набора свойств, который наверняка может быть использован при установлении соответствия пикселя той или иной области в процессе расширения. В качестве такого критерия можно использовать значения локальных максимумов и минимумов построенной гистограммы интенсивностей изображения.

Другой важной проблемой при расширении области является формулировка условия окончания процесса. Обычно процесс расширения области заканчивается, если больше не существует пикселей, удовлетворяющей критерию принадлежности к той или иной области. Для решения данной проблемы возможно применение таких критериев, как интенсивность, текстура и цвет, которые являются локальными по своей природе и не учитывают «историю» объекта. Дополнительный критерий, повышающий мощность алгоритма

расширения области, включает понятие размера, схожести между пикселем-кандидатом и присоединяющей областью.

Разбиение и объединение области. Изложенная выше процедура расширения области начинает работу с заданного множества узловых точек. Однако можно сначала разбить образ на ряд произвольных непересекающихся областей и затем объединять и/или разбивать эти области с целью удовлетворения условий, сформулированных в начале раздела. Один из способов сегментации исходного изображения состоит в разбиении площади образа на все меньшие квадратные области. Потом каждый из квадратов подвергается проверке на выполнение предиката, отвечающего за присоединение данного квадрата к образу. В качестве такого предиката может выступать величина, показывающая неразрывное заполнение проверяемого квадрата пикселями заданной интенсивности.

Итеративные алгоритмы разбиения и объединения, работа которых направлена на выполнение этих ограничений, могут быть изложены в виде процедуры, где на каждом шаге выполняются следующие операции:

1. Разбиение области R_i (на первом шаге – всего изображения), для которой выбранное условие истинности предиката не выполняется, на четыре непересекающихся квадрата.
2. Объединение соседних областей R_j и R_k , для которых $P(R_j \cup R_k) = \text{ИСТИНА}$.
3. Выход на останов, когда дальнейшее разбиение или объединение невозможно, или возврат на первый пункт.

Пример алгоритма разбиения и объединения показан на рис. 10. Это простой пример, поскольку в результате разбиения объект был полностью описан квадратами.

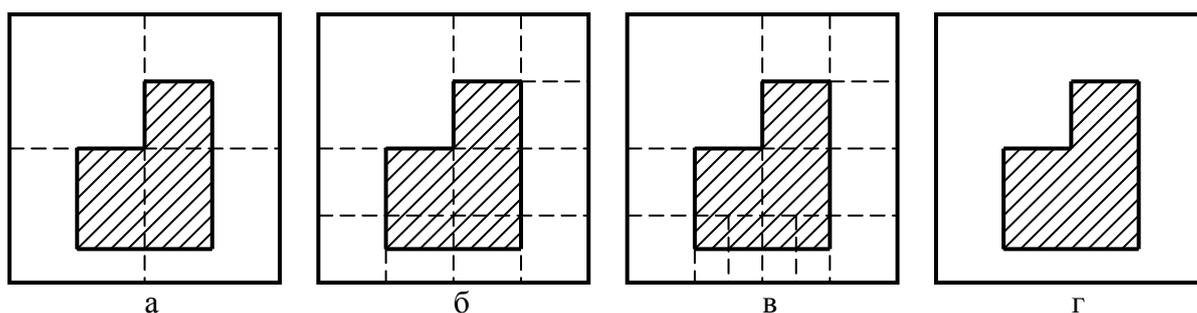


Рис. 10. Пример алгоритма разбиения и объединения.

В системах технического зрения проблемой описания называется выделение свойств (деталей) объекта с целью распознавания. В идеальном случае дескрипторы описания не должны зависеть от размеров, расположения и ориентации объекта, но должны содержать достаточное количество информации для надежной идентификации объектов. Описание является основным результатом при конструировании систем технического зрения в том смысле, что дескрипторы должны влиять не только на сложность алгоритмов распо-

знания, но также и на их работу. В рамках данной лабораторной работы рассмотрим две основные категории дескрипторов: дескрипторы границы и дескрипторы области.

3.2.1. Дескрипторы границы.

В качестве дескрипторов границ можно использовать цепные коды, которые применяются для представления границы в виде последовательности отрезков прямых линий определенной длины и направления. Обычно в основе этого представления лежит 4- или 8-связная прямоугольная решетка (см. рис. 11). Длина каждого отрезка определяется разрешением решетки, а направления задаются выбранным кодом.

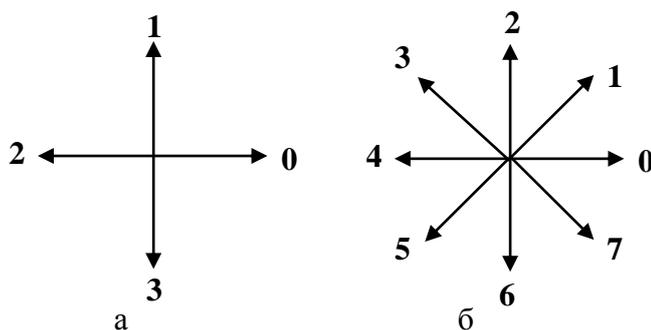


Рис. 11. 4-направленный (а) и 8-направленный цепной код (б).

Для представления всех направлений в 4-направленном цепном коде достаточно 2 бита, а для 8-направленного цепного кода требуется 3 бита. Конечно, можно определить цепные коды с большим числом направлений, но на практике наиболее часто используются коды, приведенные на рис. 4.11.

Для порождения цепного кода заданной границы исходного изображения (см. рис. 12, а) сначала выбирается решетка (см. рис. 12, б). Тогда, если площадь ячейки, расположенной внутри границы, больше определенного порогового числа (обычно 0,5), ей присваивается значение 1, в противном случае этой ячейке присваивается значение 0. Данный процесс проиллюстрирован на рис. 12, в, где ячейки изображения, площадь которых больше 0,5, отмечены темным цветом. Поскольку исходное изображение поступает на этап определения цепного кода уже дискретизованным после процедуры определения границы, то описанное преобразование не нужно.

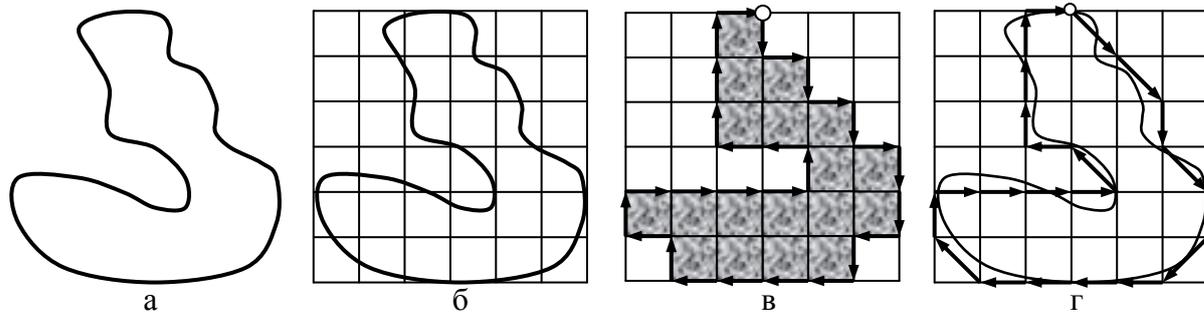


Рис. 12. Этапы получения цепного кода из исходного изображения.

После выбора начальной точки и метода кодирования цепного кода (по рис. 11), получаем описание границы контура объекта, обходя контур изображения от начальной точки по часовой стрелке. Результат описания представлен на рис. 12, *в* для 4-направленного цепного кода и на рис. 12, *г* для 8-связного кода.

Цепной код объекта на рис.4.12, *в*:

3 0 3 0 3 0 3 3 2 3 2 2 2 2 1 2 1 0 0 0 0 1 2 2 1 1 1 0.

Цепной код объекта на рис.4.12, *г*:

7 7 6 7 6 5 4 4 4 4 3 2 0 0 0 0 3 4 2 2 2 0.

Как видно, применение 8-направленного кода позволяет уменьшить его длину. Важно отметить, что цепной код зависит от выбора начальной точки и должен быть замкнутым. На этап получения цепного кода изображение поступает после процедуры поиска контура. При этом, применение 8-направленного цепного кода позволяет точнее описать границу изображения. Если для цепного кода использовать длину элементов цепи больше, чем расстояние между пикселями дискретного образа, то описание границы цепным кодом будет точнее и короче, что является большим преимуществом для систем технического зрения промышленных роботов.

3.2.2. Дескрипторы области.

Область, представляющую интерес, можно описать формой ее границы или же путем задания ее характеристик, как показано ниже. Важно отметить, что методы, рассмотренные в обоих разделах, применяются для описания областей и нормирования интересующих объектов на изображении.

Существующие СТЗ роботов основываются на довольно простых дескрипторах области, что делает их более привлекательными с вычислительной точки зрения. Однако, применение данных дескрипторов ограничено ситуациями, в которых представляющие интерес объекты различаются настолько, что для их идентификации достаточно несколько основных дескрипторов.

Площадь области определяется как число пикселей, содержащихся в пределах ее границы. Этот дескриптор полезен при сборе информации о взаимном расположении и форме объектов, от которых камера располагается приблизительно на одном и том же расстоянии. Типичным примером может служить распознавание СТЗ объектов, движущихся по конвейеру.

Большая ось объекта – это отрезок прямой линии, соединяющий две наиболее удаленные друг от друга точки объекта. При помощи данного дескриптора возможно нормирование изображения, заключающееся в расположении объекта горизонтально и увеличении его размеров до требуемого.

Большая и малая полуоси – это отрезки прямых линий, проведенные перпендикулярно к большой оси объекта в обе стороны от нее до наиболее удаленных точек, лежащих на границе объекта. Отношение длин этих осей, называемой эксцентриситетом области, также является важным дескриптором для описания формы области.

Периметром области называется длина ее границы, измеряемая в пикселях или в элементах решетки цепного кода. Чаще используется для определения *компактности* области, равной квадрату периметра, деленному на площадь.

4. Описание функций, применяемых для реализации методов СТЗ

robdiscrete – дискретизация входного изображения.

$L = \text{ROBDISCRETE}(I, N)$

$L = \text{ROBDISCRETE}(I)$

Функция **ROBDISCRETE** производит дискретизацию заданного изображения I с возможным размером пикселя 2,3,...,8. Размер пикселя задается параметром N . Входное изображение должно иметь размерность 256×256 пикселей. Если параметр N не задан, то по умолчанию используется значение, равное 2.

robquant – квантование изображения по уровням градации серого.

$L = \text{ROBQUANT}(I, N)$

$L = \text{ROBQUANT}(I)$

Функция **ROBQUANT** производит квантование изображения по уровням интенсивности. Входная матрица I должна иметь тип 'double' и размерность 256×256 пикселей. Параметром N задается количество уровней градации серого. Параметр N выбирается из интервала значений от 2 до 128. Если он не задан, то по умолчанию используется $N = 10$.

robfltmidl – фильтрация методом усреднения окрестности.

$L = \text{ROBFLTMIDL}(I, N)$

$L = \text{ROBFLTMIDL}(I)$

Функция **ROBFLTMIDL** осуществляет фильтрацию зашумленного изображения, используя алгоритм усреднения окрестности. Входное изображение I должно иметь размерность 256×256 пикселей. Параметр N задает размер области усреднения окрестности ($N \times N$) и может принимать значения 3,5,7,9,11,13. Если параметр N не задан, то по умолчанию используется $N = 3$.

robfltmidlflt – усредненная фильтрация изображения.

$L = \text{ROBFLTMIDLFLT}(I, N)$

$L = \text{ROBFLTMIDLFLT}(I)$

Функция **ROBFLTMIDLFLT** производит фильтрацию изображения методом усредненной фильтрации. Входное изображение I должно быть типа 'double' и иметь размерность 256×256 . Параметр N задает размер области усреднения. Он выбирается из ряда 3,5,7,9,11,13 и по умолчанию равен 3.

robhisttask – метод задания гистограммы.

$J = \text{ROBHISTTASK}(I, H, N)$

Функция **ROBHISTTASK** выполняет улучшение качества заданного изображения, используя метод задания гистограммы. Параметр **N** задает гистограмму в виде массива-строки, элементами которого являются плотности распределения точек заданной интенсивности. Изображение **I** необходимо предварительно проиндексировать по уровням градации серого. Параметр **N** задает количество оттенков изображения **I** (количество уровней интенсивности). Необходимо, чтобы размер массива **N** и параметр **N** совпадали. Также необходимо, чтобы сумма всех элементов **N** была равна 1.

robhistlin – метод гистограммной линеаризации изображения.

$L = \text{ROBHISTLIN}(I, N)$

Функция **ROBHISTLIN** осуществляет улучшение качества изображения методом гистограммной линеаризации. Входное изображение **I** должно быть проквантовано по уровням градаций серого (интенсивности). Параметр **N** задает количество градаций серого, по которым было проквантовано изображение.

robnoise – функция, осуществляющая зашумление изображения.

$J = \text{ROBNOISE}(I, T)$

$J = \text{ROBNOISE}(I, T, M)$

$J = \text{ROBNOISE}(I, T, M, V)$

Функция **ROBNOISE** зашумляет заданное изображение **I**. Параметром **T** задается тип шума. Если **T** = 1, то зашумление будет производиться с использованием нормально распределенного шума. В этом случае возможно использовать параметры **M** и **V**, которые задают соответственно среднее значение шума и его среднеквадратичное отклонение.

При **T**=2 на изображение накладываются случайным образом распределенные светлые и темные точки – шум типа «соль и перец». Плотность заполнения изображения шумом задается параметром **M** (параметр **V** не используется). **J** – выходное зашумленное изображение размерностью 256×256 пикселей.

Необходимо следить за тем, чтобы каждый из параметров **M** и **V** не превышал 1. По умолчанию для «нормального» шума параметр **M** = 0, а **V** = 0,01; для шума «соль и перец» **M** = 0,05. Если для «нормального» шума параметры **M** и **V** задаются, то они должны использоваться вместе.

robsegment – сегментация изображения.

$J = \text{ROBHISTTASK}(I, N)$

Функция **ROBSEGMENT** выполняет сегментацию заданного изображения. Входными параметрами функции являются изображение **I** и размерность квадрантов, на которые разбивается входное изображение.

robedge – функция поиска границы объекта.

$L = \text{ROBEDGE}(I, N)$

$L = \text{ROBEDGE}(I)$

Функция **ROBEDGE** определяет кромки заданного изображения. Входное изображение I должно быть типа 'double' и иметь размерность 256×256 пикселей. Параметр N задает шаг срабатывания градиента и выбирается из диапазона $0 \dots 3$. Если данный параметр не задан явно, то по умолчанию используется 1.

robaxes – определение большой и малой полуосей области.

$[B, S] = \text{ROBAXES}(I, V)$

Функция **ROBEDGE** определяет большую и малую оси заданного изображения. Входное изображение I должно быть типа 'double' и иметь размерность 256×256 пикселей. Выходными параметрами являются значения большой (B) и малой (S) полуосей. Дополнительно задается параметр V , значение которого определяет, отображать ли полуоси на исходном изображении. Если $V = 0$, то отображения не происходит; если $V=1$, большая полуось вычерчивается красным цветом, а малые полуоси - зеленым.

robchaincode – определение цепного кода объекта.

$J = \text{ROBCHAINCODE}(I, L, C, T)$

Входная матрица изображения I должна обладать следующим свойством: самые крайние строки и столбцы матрицы должны быть нулевыми. Параметр L задает номер строки, а C - номер столбца, с которого будет вестись отчет цепного кода, т.е. задаются координаты начальной точки. Параметр T задает первоначальное направление движения по периметру области. Выходной параметр J является искомым цепным кодом, который представляет собой вектор-строку.

robperimetr – вычисление периметра объекта.

$J = \text{ROBPERIMETR}(I)$

Функция **ROBPERIMETR** вычисляет периметр области, используя заданный цепной код (4- или 8-направленный). Входной параметр I – это цепной код области, периметр которой необходимо найти. Выходной параметр J – искомый периметр области.

robplace – вычисление площади объекта.

$J = \text{ROBPLACE}(I)$

Функция **ROBPLACE** вычисляет площадь поверхности заданной фигуры. Для того, чтобы изображение было подвергнуто процедуре определения площади, оно должно быть черно/белым, т.е. иметь 2 уровня градации серого. Площадь вычисляется как количество черных пикселей в изображении. Входной параметр I – заданное изображение, выходной параметр J – площадь искомой области.

5. Лабораторная работа №1. Методы обработки видеоинформации в системах технического зрения промышленных роботов (с использованием программного комплекса, разработанного для системы MATLAB 5.2, включая Image Processing Toolbox)

Цель работы: ознакомление и изучение основных методов обработки видеоизображений в системах технического зрения (СТЗ) промышленных роботов (ПР); приобретение навыков применения этих методов посредством моделирования процесса поэтапной обработки видеоизображений.

5.1. Порядок выполнения работы

1. Ознакомиться с теоретическим материалом, изложенным выше. Усвоить методы, приведенные в теоретическом материале, разобраться с примерами.
2. Получить у преподавателя вариант работы, а именно:
 - а) один из 10 вариантов изображений, полученных СТЗ и используемых в дальнейшем;
 - б) количество уровней градации серого изображения, с которым предстоит работать;
 - в) размерность дискретизованного изображения (128×128 , 64×64 и др.), или размер области дискретизации;
 - г) вид применяемого к изображению шума и его параметры, куда для нормально распределенного шума входят математическое ожидание и дисперсия, а для шума «соль и перец» – плотность распределения шума на изображении;
 - д) размер области сегментации изображения.

Следует помнить, что для успешного применения некоторых методов СТЗ необходимо будет подбирать предложенные параметры обработки для получения наилучших результатов. Критерием качества выбранных значений параметров обработки будет считаться получение такого изображения, на котором данный метод показал наилучшие результаты.

3. В окне «MATLAB Command Window» в командной строке запустить исполняемый файл лабораторной работы «LabRobot.m» путем задания имени файла. Появится приглашение к вводу варианта работы. Выберите вариант изображения, выданный Вам преподавателем. На экране появится окно, в котором находятся исходное цветное и черно-белое изображения предмета, с которым Вам предстоит работать. Одновременно Вам будет показано основное окно программы, в котором можно будет выбрать методы СТЗ для обработки черно-белого изображений:

1. Дискретизация изображения.

Для дискретизации исходного черно-белого изображения необходимо из списка выбрать размер области дискретизации и для применения эффекта на изображении нажать кнопку [Применить].

2. Квантование изображения по уровням градации серого (интенсивности).

Аналогично вышеприведенному – из списка выбирается количество уровней градации серого цвета. Также, в списке, расположенном над самим изображением, необходимо выбрать, какое изображение будет квантоваться: исходное черно-белое или изображение, которое уже подверглось дискретизации.

3. Зашумление и фильтрация изображения.

Область окна для задания шума на изображении разбита на четыре части: в левой верхней четверти находится панель управления; в правой верхней – исходное и зашумленное изображение; в левой нижней – изображение, обработанное по методу усреднения окрестности; в правой нижней – изображение, обработанное по методу усредненной фильтрации. Необходимо выбрать, над каким изображением будет произведено преобразование. Это можно сделать в списке, расположенном над исходным изображением.

Для правильного выполнения данного пункта лабораторной работы необходимо «зашумить» изображение. Для этого нужно выбрать тип применяемого шума и его параметры, заданные преподавателем. Выбрав параметры шума, необходимо нажать кнопку [Применить], чтобы применить заданные параметры к исходному изображению. Зашумленное изображение появится в правой верхней четверти вместо исходного. Затем необходимо обработать уже зашумленное изображение с использованием методов усреднения окрестности и усредненной фильтрации. Для каждого метода необходимо в выпадающем меню выбрать размер области усреднения, и нажать кнопку [Применить]. Успешное выполнение данного пункта лабораторной работы заключается в подборе таких значений применяемой области усреднения для каждого метода, чтобы результаты использование метода были наиболее эффективными в смысле устранения шума.

4. Работа с гистограммой.

Как и в предыдущих случаях, для начала нужно выбрать изображение, подвергающееся в дальнейшем обработке. В методе задания гистограммы прежде всего необходимо задать гистограмму. После этого нажать кнопку [Применить], расположенную под изображением с надписью «Метод задания гистограммы», а также кнопку, расположенную под изображением с надписью «Гистограммная линейаризация».

При изменении гистограммы процедура обработки сохраняется: сначала обработка по методу задания гистограммы, затем обработка по методу гистограммной линеаризации.

5. Поиск границы объекта.

Предварительно нужно выбрать изображение из предложенного списка. Необходимо в строке редактирования задать шаг срабатывания градиента (0...3) и нажать кнопку [Применить]. Выбранное за исходное изображение будет подвергнуто процедурам вычисления градиента по осям O_x и O_y . Итоговое изображение после применения метода поиска границы объекта будет получено в результате объединения изображений, содержащих градиенты границы объекта. Все три изображения будут продемонстрированы в соответствующем окне программы. Процедура поиска границы заканчивается, если путем подбора шага срабатывания градиента итоговое изображение представляет собой четкую одинарную границу объекта, желательно без разрывов.

6. Определение цепного кода, периметра и площади объекта.

На изображении крестиком показана точка, с которой начинается проведение цепного кода. Его элементы можно просмотреть в командном окне MATLAB; также там указаны координаты этой точки, площадь и периметр объекта.

7. Определение основной оси, большой и малой полуосей.

В появившемся окне нажать кнопку [Применить]. После этого на изображении появиться основная (красного цвета) ось, и большая и малая (зеленого цвета) полуоси. Численные значения большой и малой полуосей выводятся в командном окне MATLAB.

8. Сегментация изображения.

Для сегментации необходимо указать размер области сегментации из списка по варианту, предложенному преподавателем, а затем нажать кнопку [Применить]. Появится черный профиль объекта.

4. Фиксировать результаты работы программы и делать выводы по проделанной работе в результате моделирования методов СТЗ.

5. Представить отчет о работе.

Примечания.

- Работа каждого метода может занимать некоторое время (в пределах 1-2 минут).
- В верхней части некоторых окон присутствует список, из которого можно выбрать изображение, над которым будет производиться работа; поэтому перед тем, как нажимать кнопку [Применить], необходимо сделать выбор изображения из списка.

5.2. Пример выполнения работы

Пусть, к примеру, задан седьмой вариант. При запуске в командном окне MATLAB Command Window программы LabRobot появиться окно с заголовком «Выбор варианта работы», в котором необходимо выбрать седьмой вариант из списка и нажать кнопку [Выбрать]. После этого это окно автоматически закроется и появятся два новых окна: «Исходное цветное и черно-белое изображение» и окно с заголовком «Лабораторная работа (Вариант № 7)».

Окно с заголовком «Лабораторная работа (Вариант № 7)» является основным. В нем присутствуют три элемента управления ходом работы: кнопка «Закрыть все окна и выйти из программы», при нажатии которой работа прекращается; список, в котором выбираются различные методы работы с изображением (черно-белым); кнопка [Применить], нажатие которой приводит к активации соответствующего элемента из списка.

Прежде всего, необходимо произвести дискретизацию изображения. Для этого из списка основного окна необходимо выбрать пункт «1. Дискретизация изображения» и нажать кнопку [Применить]. После этого появиться окно с заголовком «Дискретизация изображения», в котором присутствует список, из которого выбирается размер области дискретизации. Кнопка [Применить] этого окна позволяет произвести дискретизацию выбранного изображения и вывести дискретизованное изображение на монитор.

После этого производится квантование изображения по уровням интенсивности. Так же как и в предыдущем методе, из списка необходимо выбрать количество уровней градаций серого цвета, с помощью которых будет представлено исходное изображение, и нажать кнопку [Применить]. Из списка вверху окна необходимо выбрать дискретизованное изображение.

Затем необходимо изучить методы фильтрации изображения. Для этого в основном окне программы из списка выбирается пункт «3. Зашумление и фильтрация изображения». Появится окно с аналогичным заголовком. Допустим, по условия задания необходимо провести зашумление изображения нормально распределенным шумом с параметрами: математическое ожидание равно 0,01, дисперсия шума составляет 0,01. Для этого в поле ввода «Мат. ож.», соответствующего нормально распределенному шуму, вводим значение 0,01, а в поле ввода «Дисперс.» выставляем значение 0,01. Теперь необходимо нажать кнопку [Применить] соответствующей вкладки. Справа вверху появится зашумленное изображение.

После этого необходимо промоделировать работу методов фильтрации изображения. При использовании метода усреднения окрестности в выпадающем меню под надписью «Размер области усреднения для метода усреднения окрестности» выбирается размер области усреднения. По нажатию кнопки [Применить], расположенной справа от этого списка, производится запуск процедуры усреднения и вывод полученного изображения на экран. Аналогичным образом производится работа с методом усредненной филь-

трации. Завершение работы с методами усреднения производится после подбора таких значений области усреднения для каждого метода, при применении которых к исходному изображению получаются самые качественные результаты.

Работа с гистограммой осуществляется после выбора в основном окне программы пункта «4. Работа с гистограммой» – появится окно с одноименным заголовком. В поле ввода, расположенном справа, вводится заданная гистограмма в формате записи вектора-строки. Количество элементов в этом векторе определяет количество используемых уровней градации серого цвета. Пусть необходимо выровнять изображение по следующей гистограмме: [0,1 0,3 0,2 0,4] (размер данного вектора 4×1 , следовательно, количество градаций серого цвета равно 4). Для этого данную гистограмму вводим в поле ввода и нажимаем кнопку [Применить], расположенную под изображением с названием «Метод задания гистограммы». В результате выполнится гистограммное выравнивание по введенной гистограмме. Затем проверяется работа метода гистограммной линейаризации. Для этого необходимо просто нажать кнопку [Применить], расположенную под изображением с надписью «Гистограммная линейаризация».

После работы с гистограммой необходимо найти границы изображения. В основном окне из списка выбирается пункт «5. Поиск границы объекта», после чего появляется окно с одноименным заголовком. В поле ввода «Срабатывание градиента» вносится значение величины срабатывания градиента (от 0 до 3). Чем больше данная величина, тем большее количество вторичных контуров будет представлено на обработанном изображении. Но, при достаточно малой величине уровня срабатывания градиента контур объекта будет прерывистым. После ввода значения величины срабатывания градиента необходимо нажать кнопку [Применить], после чего будут найдены и представлены на рисунках градиент изображения по осям Ox и Oy появится изображение границ объекта (белым на черном).

Определение цепного кода производится следующим образом. В основном окне программы из списка выбирается пункт «6. Нахождение цепного кода, периметра и площади объекта», после чего появляется окно с одноименным заголовком. На представленном изображении крестиком отмечена точка, с которой пойдет образовываться цепной код. Нажатие кнопки [Применить] в этом окне приводит к выполнению процедуры нахождения цепного кода, первые 20 элементов которого выводится в командное окно MATLAB Command Window. Туда же выводятся координаты начальной точки и первоначальное направление для цепного кода, а также периметр и площадь объекта.

Предпоследним этапом выполнения работы является определение большой и малой полуосей объекта. Аналогично вышеприведенным шагам в главном окне выбирается предпоследний пункт списка – «7. Определение большой и малой полуосей». В появившемся окне необходимо нажать кнопку [Применить], после чего на изображении появятся полуоси.

И, наконец, последний шаг – проведение сегментации изображения. В главном окне выбирается пункт «8. Сегментация изображения», и в появившемся окне из списка выбирается размер области сегментации, после чего необходимо нажать кнопку [Применить] для проявления эффекта.