

ция принадлежности просто задает порог для возраста, при котором (и меньших значениях) человек считается молодым, а выше которого — немолодым. Предположим, что мы определили как молодых всех людей, имеющих возраст 20 лет и меньше. Сразу появляется трудность: человек, чей возраст составляет 20 лет и 1 секунду, не должен входить в множество молодых людей. Такое ограничение возникает вне зависимости от значения порога, принятого для классификации людей как молодых. Хотелось бы иметь более гибкое определение того, что подразумевается под «молодостью», т. е. иметь *постепенный* переход из молодых в немолодые. Теория *нечетких множеств* реализует эту идею с помощью функций принадлежности, которые постепенно меняются в интервале между граничными значениями 1 (определенно молодой) и 0 (определенно немолодой). С помощью нечетких множеств можно делать утверждение, что некоторый человек на 50 % молодой (находится в середине перехода от молодого к немолодому). Иначе говоря, возраст представляет собой неточное понятие, и нечеткая логика дает возможность работы с такими понятиями. Нечеткие множества будут подробнее рассматриваться в разделе 3.8.

2.6.5. Пространственные операции

Пространственные операции выполняются непосредственно над значениями пикселей данного изображения. Эти операции можно разделить на три широких класса: (1) поэлементные операции (поточечная обработка), (2) операции над окрестностью (локальная обработка) и (3) геометрические преобразования (глобальная обработка).

Поэлементные операции

Простейшая операция над цифровым изображением состоит в изменении значений отдельных пикселей исходя из их яркости. Такой способ обработки может быть выражен *функцией градационного преобразования* (также называемой *функцией преобразования интенсивностей* или *функцией отображения*) вида

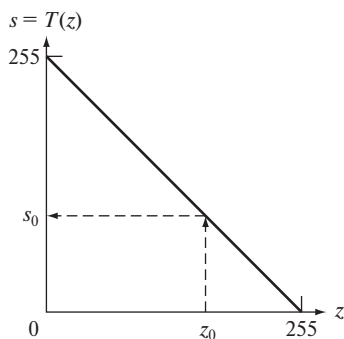


Рис. 2.34. Функция градационного преобразования, используемая для получения негатива 8-битового изображения. Пунктирные стрелки показывают, как произвольное значение исходной яркости z_0 преобразуется в значение яркости s_0 соответствующего пикселя на результирующем изображении

$$s = T(z), \quad (2.6-20)$$

где z — яркость пикселя исходного изображения, а s — полученная после преобразования яркость соответственного пикселя на обработанном изображении. Например, на рис. 2.34 изображена функция преобразования, используемая для получения негатива 8-битового изображения, такого как на рис. 2.32(б), который был получен с помощью операций над множествами. В главе 3 рассматривается ряд методов задания функций градиационного преобразования.

Операции над окрестностью

Обозначим S_{xy} множество координат окрестности произвольной точки (x, y) изображения f . При обработке окрестности генерируется соответственный (имеющий те же координаты) пиксель в выходном изображении g таким образом, что его значение определяется с помощью заданной операции над пикселями исходного изображения с координатами из S_{xy} . Например, пусть эта операция состоит в вычислении среднего значения яркости пикселей прямоугольной окрестности с размерами $m \times n$ и центром в точке (x, y) . Координаты пикселей в этой области составляют множество S_{xy} . Рис. 2.35(а) и (б) иллюстрируют такой процесс. Эту операцию можно выразить уравнением

$$g(x, y) = \frac{1}{mn} \sum_{(r, c) \in S_{xy}} f(r, c), \quad (2.6-21)$$

где r и c суть координаты строки и столбца для тех пикселей, координаты которых входят в множество S_{xy} . Изображение g формируется путем варьирования координат (x, y) так, что центр окрестности передвигается по изображению f от пикселя к пикселю и в каждом новом месте повторяется та же самая операция над окрестностью. Например, изображение на рис. 2.35(г) получено таким способом с помощью окрестности размерами 41×41 . Достижимый эффект выражается в локальной расфокусировке исходного изображения. Такой способ обработки применяется, например, для устранения мелких деталей и тем самым выявления «пятен», соответствующих самым крупным областям изображения. Локальная обработка рассматривается в главах 3 и 5, а также в других местах книги.

Геометрические преобразования и совмещение изображений

Геометрические преобразования изменяют пространственные взаимосвязи между пикселями на изображении. Геометрические преобразования часто называют *преобразованиями резинового листа*, поскольку их можно представить себе как процесс распечатки изображения на листе из резины и дальнейшее растягивание этого листа в соответствии с определенными правилами. С точки зрения цифровой обработки изображений геометрические преобразования состоят из следующих двух основных операций: (1) пространственное преобразование координат и (2) *интерполяция значений яркости*, при которой происходит присвоение значений яркости точкам изображения, подвергнутого пространственному преобразованию.

Преобразование координат может быть выражено в форме уравнения

$$(x, y) = T\{(v, w)\}, \quad (2.6-22)$$

а б
в г

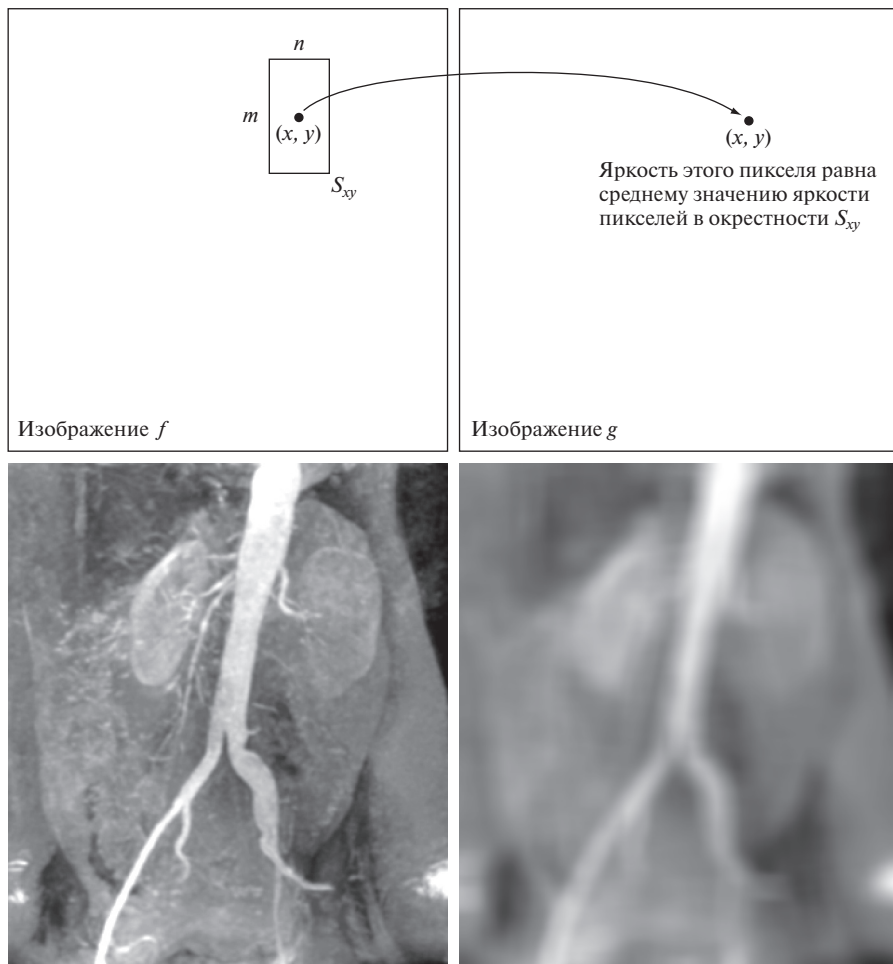
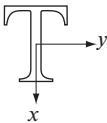
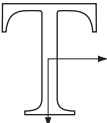
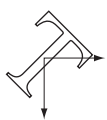
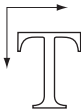
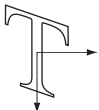
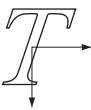


Рис. 2.35. Локальное усреднение с помощью обработки окрестности. Процедура для прямоугольной окрестности показана на рисунках (а) и (б). (в) Ангиограмма аорты из раздела 1.3.2. (г) Результат использования уравнения (2.6-21) с $m = n = 41$. Размеры изображений 790×686 пикселей

где (v, w) — координаты пикселя на исходном изображении, а (x, y) — координаты соответственного пикселя на преобразованном изображении. Например, преобразование $(x, y) = T\{(v, w)\} = (v/2, w/2)$ сжимает исходное изображение до половины первоначального размера в каждом направлении. Одним из наиболее часто применяемых преобразований координат является *аффинное преобразование* [Wolberg, 1990], которое имеет общую форму

$$[x \ y \ 1] = [v \ w \ 1] \mathbf{T} = [v \ w \ 1] \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}. \quad (2.6-23)$$

Таблица 2.2. Аффинные преобразования на основе уравнения (2.6-23)

Название преобразования	Аффинная матрица T	Преобразование координат	Пример
Тождественное преобразование	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Изменение масштаба	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Поворот	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\sin\theta + w\cos\theta$	
Параллельный перенос (сдвиг)	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Вертикальный скос	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Горизонтальный скос	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

В зависимости от выбранных значений элементов матрицы T , это преобразование позволяет осуществлять изменение масштаба, поворот, сдвиг или скос множества координатных точек. В табл. 2.2 указаны значения элементов матрицы, используемые для реализации этих преобразований. Реальная сила матричного представления (2.6-23) состоит в том, что оно предлагает единый каркас для выполнения последовательности различных операций. Например, чтобы изменить размеры изображения, повернуть его и передвинуть результат в другое место, достаточно просто сформировать матрицу 3×3 , равную произведению матриц масштабирования, поворота и параллельного переноса из табл. 2.2.

Вышеописанные преобразования перемещают пиксели изображения в новые места. Для завершения процесса необходимо присвоить им значения яркости. Эта задача решается с помощью интерполяции яркостей, которая уже рассматривалась в разделе 2.4.4. Там мы начали с примера, связанного с увеличением изображения, и рассмотрели вопрос о присвоении значений яркости пикселей на новых местах. Увеличение — это просто масштабное преобразование, указанное во второй строке табл. 2.2, и для других преобразований в табл. 2.2 применимы аналогичные способы присвоения значений яркости перемещен-

ным пикселям, как и те, что рассматривались для увеличения. Как и в разделе 2.4.4, при выполнении этих преобразований можно использовать интерполяцию по ближайшему соседу, билинейную и бикубическую интерполяции.

На практике уравнение (2.6-23) можно использовать двумя способами. Первый, который называется *прямое отображение*, состоит в сканировании пикселей исходного изображения с вычислением в каждой координатной точке (v, w) пространственных координат (x, y) соответственного пикселя на выходном изображении, прямо применяя уравнение (2.6-23). С подходом, основанным на прямом отображении, связана та проблема, что два и более пикселей исходного изображения могут преобразоваться в одно и то же место на выходном изображении, так что встает вопрос о том, как совместить несколько полученных значений в одном выходном пикселе. Кроме того, может оказаться, что некоторым пикселям выходного изображения не будет присвоено вообще никакого значения. При втором подходе, называемом *обратным отображением*, сканируются координаты пикселей выходного изображения и для каждой точки (x, y) вычисляется соответственная позиция в исходном изображении с помощью соотношения $(u, v) = T^{-1}(x, y)$. Затем в найденной точке выполняется интерполяция одним из способов, описанных в разделе 2.4.4, с использованием значений соседних пикселей исходного изображения, в результате чего находится яркость очередного пикселя на выходном изображении. Обратное отображение реализуется более эффективно, чем прямое, и используется в многочисленных коммерческих реализациях пространственных преобразований (например, этот подход используется в программном пакете MATLAB).

Пример 2.9. Поворот изображения с интерполяцией яркости.

■ Целью этого примера является проиллюстрировать поворот изображения с помощью аффинного преобразования. На рис. 2.36(а) показано исходное изображение с разрешением 300 dpi, а рис. 2.36(б)—(г) демонстрируют результаты поворота этого изображения на 21° с интерполяцией яркости по ближайшему соседу, билинейной и бикубической интерполяцией соответственно. Поворот

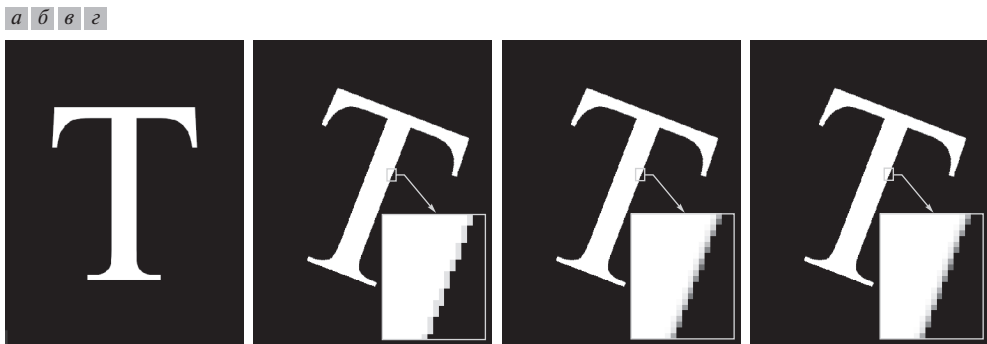


Рис. 2.36. (а) Изображение буквы Т с разрешением 300 dpi. (б) Результат поворота на 21° с использованием интерполяции по ближайшему соседу для присвоения значений яркости преобразованным пикселям. (в) То же при использовании билинейной интерполяции. (г) То же при использовании бикубической интерполяции. Увеличенные фрагменты показывают детали перепада для трех интерполяционных методов

является одним из наиболее требовательных геометрических преобразований в плане сохранения прямолинейности объектов, присутствующих в исходном изображении. Как видно из рисунка, интерполяция по ближайшему соседу приводит к появлению наиболее ступенчатых краев, а билинейная интерполяция существенно улучшает результат, как отмечалось в разделе 2.4.4. Как и прежде, использование бикубической интерполяции немного повышает резкость. В самом деле, при сравнении увеличенных фрагментов на рис. 2.36(в) и (г) видно, что в середине фрагмента число вертикальных серых «блоков» в области перехода яркости от светлого к темному на рис. 2.36(в) больше, чем в соответственной области (г), что говорит о более высокой резкости перепада на последнем изображении. Сходные результаты можно получить и при других пространственных преобразованиях из табл. 2.2, которые требуют интерполяции (тождественное преобразование не требует, как и параллельный перенос, если величина сдвига по каждой координатной оси выражается целым числом пикселей). В этом примере использовалось обратное отображение, рассмотренное в предыдущем абзаце. ■

Важной прикладной задачей цифровой обработки изображений является *совмещение изображений*, т.е. взаимное выравнивание двух или более изображений одной и той же сцены. В рассмотрении выше мы считали известным вид функции, необходимой для получения желаемого результата преобразования исходного изображения. При совмещении изображений у нас помимо исходного есть еще преобразованное изображение, но в общем случае неизвестно конкретное преобразование, которое привело к такому результату. Задача состоит в том, чтобы по двум изображениям оценить функцию преобразования и затем воспользоваться ей для совмещения этих изображений. Для уточнения терминологии будем называть исходным то изображение, которое мы хотим преобразовать, а *эталонным* изображением назовем то, с которым хотим совместить исходное изображение.

Например, может представлять интерес совмещение двух или более изображений, полученных приблизительно в одно и то же время с помощью двух разных систем медицинской интроскопии, например ЯМР (томография на основе ядерного магнитного резонанса) и ПЭТ (позитронная эмиссионная томография) или изображения, полученные одним и тем же прибором в разное время, например спутниковые изображения интересующего участка местности, снятые с интервалом в несколько дней, месяцев или даже лет. В любом случае для комбинирования изображений или проведения их сравнения и количественного анализа необходимо компенсировать геометрические искажения, возникшие из-за различий угла наблюдения, расстояния и ориентации регистрирующей системы, разрешения сенсора, изменения положения объекта и прочих факторов.

Один из главных подходов к решению изложенной задачи состоит в использовании *точек привязки*, которые представляют собой такое подмножество соответственных пикселей, положение которых на исходном и эталонном изображениях точно известно. Для выбора системы точек привязки существуют различные методики, ориентированные на конкретные приложения, от интерактивного выбора точек до использования алгоритмов автоматического обнаружения таких точек. Так, некоторые системы формирования изображений

обладают характерными особенностями, имеющими физическую природу, такими как специфические элементы конструкции самого сенсора (например небольшие металлические метки на нем). В результате прямо на изображении в процессе его формирования возникает множество точек с известными координатами (эти точки называют *опорными* или точками *ризо-маркировки*), которые служат основой системы точек привязки.

Задача оценивания функции преобразования — это задача моделирования. Например, предположим, что на исходном и эталонном изображениях имеется по четыре точки привязки. Простая модель преобразования, основанная на билинейной аппроксимации, задается выражениями:

$$x = c_1 v + c_2 w + c_3 vw + c_4 \quad (2.6-24)$$

$$\text{и} \quad y = c_5 v + c_6 w + c_7 vw + c_8, \quad (2.6-25)$$

где на этапе оценивания (v, w) и (x, y) — координаты точек привязки в исходном и эталонном изображениях соответственно. Поскольку имеется четыре пары соответственных точек привязки в двух изображениях, можно выписать восемь уравнений (2.6-24) и (2.6-25), а затем решить полученную систему уравнений относительно коэффициентов $c_i, i = 1, 2, \dots, 8$. Эти коэффициенты задают модель геометрических преобразований всех пикселей одного изображения в такие координаты пикселей на другом изображении, при которых обеспечивается совмещение изображений.

Как только коэффициенты получены, уравнения (2.6-24) и (2.6-25) можно использовать для преобразования всех пикселей исходного изображения в пиксели нового изображения, которое должно совместиться с эталонным при условии правильного выбора точек привязки⁸. В ситуациях, когда для удовлетворительного совмещения изображений четырех точек недостаточно, часто применяемый метод состоит в выборе большего числа точек привязки и использовании четырехугольников, образованных четверками соседних точек, в качестве более мелких изображений. Эти субизображения совмещаются, как описано выше, так что все пиксели внутри четырехугольника преобразуются с помощью коэффициентов, найденных на основании четырех точек привязки, являющихся его вершинами. Затем та же процедура повторяется для следующей четверки точек привязки и так далее, пока не будут обработаны все четырехугольники. Разумеется, могут применяться области более сложной формы, чем четырехугольные, и более сложные модели функции преобразования, например полиномиальные с нахождением приближения методом наименьших квадратов. В общем случае число требуемых точек привязки и сложность модели, которые необходимы для удовлетворительного решения задачи, зависят от степени серьезности геометрического искажения. Наконец, необходимо помнить, что преобразования вида (2.6-24) и (2.6-25) или любые другие модели для той же цели задают только отображение пространственных координат пикселей на исходном изображении. Необходимо еще провести интерполяцию яркости пикселей на результирующем изображении с помощью любого из рассмотренных выше методов.

⁸ И, что не менее важно, при правильно выбранной модели искажения. — *Прим. перев.*

Пример 2.10. Совмещение изображений.

■ На рис. 2.37(а) показано эталонное изображение, а рис. 2.37 демонстрирует то же изображение после геометрического искажения в форме скоса по вертикали и по горизонтали. Наша цель — выбрать точки привязки на эталонном (и искаженном) изображениях и по ним совместить изображения. Выбранные (вручную) точки привязки показаны в виде маленьких белых квадратиков вблизи углов изображений (в данном случае нам достаточно всего четырех точек привязки, поскольку искажающий оператор есть линейный скос в обоих направлениях). На рис. 2.37(в) показан результат привязки исходного изображения по этим точкам с помощью процедуры, изложенной в предыдущем

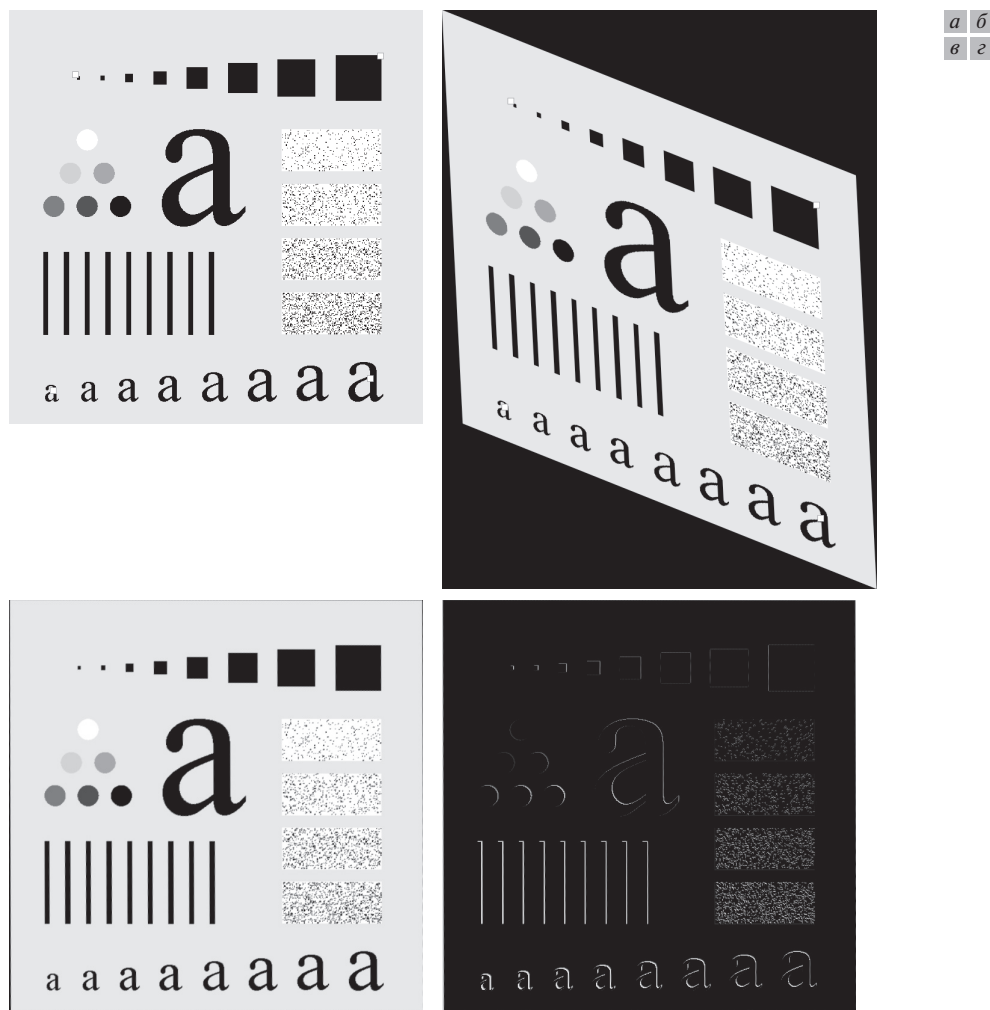


Рис. 2.37. Совмещение изображений. (а) Эталонное изображение. (б) Исходное (геометрически искаженное) изображение. Соответственные точки привязки указаны маленькими белыми квадратиками вблизи углов. (в) Изображение после совмещения (видны дефекты на краях). (г) Разность изображений (а) и (в) показывает другие ошибки совмещения

абзаце. Видно, что достигнутое совмещение не идеально, о чем свидетельствуют черные края на рисунке. Разностное изображение на рис. 2.37(г) более ясно демонстрирует легкие несовпадения между эталонным и откорректированным изображениями. Причиной рассогласования является неточность при ручном выборе точек привязки. Трудно достичь идеального совмещения, когда искажения столь значительны. ■

2.6.6. Векторные и матричные операции

Обработка многоканальных изображений является типичной областью использования векторных и матричных операций над изображениями. Например, из главы 6 мы знаем, что цветные цифровые изображения формируются в цветовом пространстве RGB с помощью трех изображений красной, зеленой и синей цветовых компонент, как показано на рис. 2.38. Как мы видим, здесь *каждый* пиксель цветного изображения имеет три компоненты, которые можно записать в форме *вектор-столбца*

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}, \quad (2.6-26)$$

где z_1 есть яркость пикселя на красном изображении, а другие два элемента суть яркости соответственных пикселей на зеленом и синем изображении. Таким образом, цветное RGB-изображение размерами $M \times N$ можно представить тремя изображениями-компонентами тех же размеров или совокупностью MN трехмерных векторов. В общем случае многоканальное изображение, состоящее из n компонент (см., например, рис. 1.10), приводит к n -мерным векторам. Такое векторное представление будет местами использоваться в главах 6, 10, 11 и 12.

В разделе «Обучающие материалы» на сайте книги в Интернете имеется краткое введение в теорию векторов и матриц.

Представив пиксели в форме векторов, мы получаем в распоряжение аппарат линейной алгебры и аналитической геометрии. Например, *евклидово рас-*

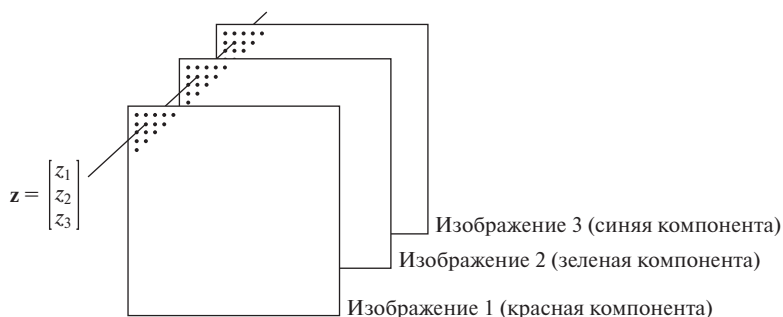


Рис. 2.38. Построение вектора из значений соответственных пикселей трех компонент RGB изображения

стояние D между векторным пикселем \mathbf{z} и произвольной точкой \mathbf{a} в n -мерном пространстве определяется как скалярное произведение векторов:

$$D(\mathbf{z}, \mathbf{a}) = [(\mathbf{z} - \mathbf{a})^T (\mathbf{z} - \mathbf{a})]^{\frac{1}{2}} = [(z_1 - a_1)^2 + (z_2 - a_2)^2 + \dots + (z_n - a_n)^2]^{\frac{1}{2}}, \quad (2.6-27)$$

что является обобщением евклидова расстояния на плоскости, определенного в (2.5-1). Правую часть в (2.6-27) иногда называют *нормой вектора* и обозначают $\|\mathbf{z} - \mathbf{a}\|$. В дальнейшем мы неоднократно будем вычислять это расстояние.

Другое важное преимущество векторного представления пикселей возникает в линейных преобразованиях, которые приобретают простой вид

$$\mathbf{w} = \mathbf{A}(\mathbf{z} - \mathbf{a}), \quad (2.6-28)$$

где \mathbf{A} есть матрица порядка $m \times n$, а \mathbf{z} и \mathbf{a} суть вектор-столбцы размерности $n \times 1$. Как мы увидим в дальнейшем, преобразования такого типа в ряде случаев полезны при обработке изображений.

Как отмечалось в соотношении (2.4-2), изображение целиком может рассматриваться как матрица (или эквивалентно как вектор), что имеет важные последствия для решения многих задач обработки изображений. Например, можно представить изображение с размерами $M \times N$ в виде вектора размерности $MN \times 1$, используя первую строку изображения в качестве первых N элементов этого вектора, вторую строку — как следующие N элементов и т. д. Если изображение представлено таким способом, то широкий круг линейных преобразований, применяемых к изображению, можно описать в единой форме

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \quad (2.6-29)$$

где вектор \mathbf{f} размерности $MN \times 1$ представляет исходное изображение, вектор \mathbf{n} размерности $MN \times 1$ представляет шумовую составляющую размерами $M \times N$, вектор \mathbf{g} размерности $MN \times 1$ представляет обработанное изображение, а \mathbf{H} — матрица порядка $MN \times MN$, которая задает линейное преобразование, применяемое к исходному изображению (см. раздел 2.6.2 о линейных преобразованиях). Например, начав с (2.6-29), можно разработать всю совокупность обобщенных методов восстановления изображений, как мы увидим в разделе 5.9. Мы коснемся темы использования матричного представления в следующем разделе и продемонстрируем другие применения матриц при цифровой обработке изображений в главах 5, 8, 11 и 12.

2.6.7. Преобразования изображений

Все до сих пор обсуждавшиеся методы обработки цифровых изображений оперировали непосредственно пикселями исходного изображения, т. е. работали прямо в *пространственной области*. В некоторых случаях задачи обработки изображений легче формулируются после преобразования исходного изображения, которое переводит задачу в *область преобразования*, а для возвращения обратно в пространственную область используется обратное преобразование. По мере освоения книги мы встретимся с разнообразными преобразованиями. Особенно важный класс двумерных линейных преобразований, которые будем обозначать $T(u, v)$, в общем виде описывается выражением