$$\begin{array}{ccc} & p_3 & p_4 \\ p_1 & p_2 & \\ p & & \end{array}$$

Предположим, что рассматривается смежность пикселей со значением 1, т.е. $V = \{1\}$. Если оба элемента p_1 и p_3 имеют значения 0, то длина кратчайшего m-пути (т.е. расстояние D_m) между p и p_4 равна 2. Если значение p_1 равно 1, то элементы p и p_2 больше не являются m-смежными (см. определение отношения m-смежности) и длина кратчайшего m-пути становится равной 3 (этот путь проходит через точки p, p_1, p_2, p_4). Аналогичные рассуждения имеют место в том случае, если значение p_3 равно 1 (а значение p_1 равно 0). В этом случае длина кратчайшего m-пути также равна 3. Наконец, если оба пикселя p_1 и p_3 имеют единичные значения, то длина кратчайшего m-пути между p и p_4 станет равной 4. В таком случае путь проходит через последовательность точек p, p_1, p_2, p_3, p_4.

2.6. Введение в математический аппарат, применяемый в цифровой обработке изображений

Этот раздел преследует две главные цели: (1) познакомить читателя с различными математическими инструментами, которые используются на протяжении всей книги и (2) помочь читателю ощутить, как именно этот аппарат используется, применяя его в разнообразных простых задачах обработки изображений, часть которых будет неоднократно появляться в дальнейших обсуждениях. По мере необходимости область применения этих инструментов будет расширяться в последующих главах.

> Перед тем, как двигаться дальше, читателю может быть полезно загрузить и изучить обзорный материал из раздела «Обучающие материалы» на сайте книги в Интернете. Этот обзор содержит вводный материал о матрицах и векторах, линейных системах, теории множеств и теории вероятности.

2.6.1. Поэлементные и матричные операции

Поэлементные операции, в которых участвуют одно или более изображений, всегда выполняются *попиксельно* над соответственными элементами изображений. Ранее в этой главе упоминалось, что изображения можно также эквивалентно рассматривать как матрицы. И в самом деле, во многих случаях операции над изображениями выполняются по правилам матричной алгебры (см. раздел 2.6.6). Именно по этой причине необходимо четко разграничить поэлементные и матричные операции. Рассмотрим, например, следующие два изображения размерами 2×2:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ и } \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

Поэлементное произведение этих двух изображений вычисляется так:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}.$$

Напротив, *матричное произведение* изображений определяется выражением

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{bmatrix}.$$

В последующем на протяжении книги мы всюду подразумеваем, что операции выполняются попиксельно, если не оговорено иное. Например, говоря о возведении цифрового изображения в степень, мы имеем в виду, что значение каждого пикселя по отдельности возводится в эту степень; говоря об операции деления одного изображения на другое, мы на самом деле подразумеваем, что деление выполняется для соответственных пикселей двух изображений и т. д.

2.6.2. Линейные и нелинейные преобразования

Одна из важнейших характеристик любого метода обработки изображений — это является ли он линейным или нелинейным. Рассмотрим оператор общего вида H, который строит выходное изображение $g(x, y)$ для данного входного изображения $f(x, y)$:

$$H\left[f(x,y)\right] = g(x,y). \tag{2.6-1}$$

Говорят, что оператор H *линейный*, если

$$H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] = a_i H\left[f_i(x,y)\right] + a_j H\left[f_j(x,y)\right] = a_i g_i(x,y) + a_j g_j(x,y), \tag{2.6-2}$$

где $f_i(x, y)$ и $f_j(x, y)$ — любые изображения одинаковых размеров, а a_i и a_j — произвольные константы. Соотношение (2.6-2) показывает, что результат применения линейного оператора к сумме двух входных изображений совпадает с суммой результатов применения такого оператора к этим изображениям по отдельности. А также результат применения линейного оператора к изображению, умноженному на константу, идентичен умножению на эту константу результата применения оператора к исходному изображению. Первое свойство называется *аддитивностью*, а второе — *однородностью.*

Рассмотрим простой пример, где в качестве H используется оператор суммы Σ, функция которого состоит просто в суммировании его входов. Для проверки линейности этого оператора начнем с левой части (2.6-2) и попытаемся доказать, что она равна правой части:

$$\begin{aligned} \sum\left[a_i f_i(x,y) + a_j f_j(x,y)\right] &= \sum a_i f_i(x,y) + \sum a_j f_j(x,y) = \\ &= a_i \sum f_i(x,y) + a_j \sum f_j(x,y) = a_i g_i(x,y) + a_j g_j(x,y), \end{aligned}$$

где первый шаг преобразования вытекает из дистрибутивности сложения. Таким образом, левая часть (2.6-2) равна правой, и можно заключить, что оператор суммы линейный.

Здесь всюду используется поэлементное суммирование, а не сумма всех пикселей изображения. Применение оператора суммы к одному изображению дает само это изображение.

Напротив, рассмотрим оператор max, функция которого состоит в нахождении максимального значения пикселя во входном изображении. Для наших целей простейший способ доказать, что этот оператор нелинейный, — найти контрпример, для которого нарушается условие (2.6-2). Рассмотрим следующие два изображения:

$$f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \text{ и } f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix},$$

и предположим, что константы $a_1 = 1$ и $a_2 = -1$. Проверку линейности опять начнем с левой части (2.6-2):

$$\max\left\{(1)\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = \max\left\{\begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix}\right\} = -2.$$

Действуя теперь с правой частью, получим

$$(1)\max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)\max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = 3 + (-1)7 = -4.$$

В данном случае левая и правая части (2.6-2) не равны друг другу, и тем самым доказано, что в общем случае оператор max является нелинейным.

Как мы увидим в трех следующих главах, особенно в главах 4 и 5, линейные операторы исключительно важны для обработки изображений, поскольку они опираются на значительную совокупность хорошо изученных теоретических и практических результатов. Нелинейные операторы исследованы значительно хуже, поэтому область их применения более ограничена. Однако мы познакомимся в последующих главах с несколькими нелинейными операциями обработки изображений, результаты которых значительно превосходят те, которые достигаются с помощью линейных аналогов.

2.6.3. Арифметические операции

Арифметические операции над изображениями являются поэлементными операциями, т. е., как уже говорилось в разделе 2.6.1, они применяются к паре соответственных пикселей двух изображений. Эти четыре арифметические операции обозначаются следующим образом:

$$\begin{aligned} s(x,y) &= f(x,y) + g(x,y), \\ d(x,y) &= f(x,y) - g(x,y), \\ p(x,y) &= f(x,y) \times g(x,y), \\ v(x,y) &= f(x,y) \div g(x,y). \end{aligned} \tag{2.6-3}$$

Понятно, что эти операции применяются к соответственным парам элементов изображений f и g для $x = 0, 1, 2, \ldots, M-1$ и $y = 0, 1, 2, \ldots, N-1$, где, как обычно, M и N — число строк и столбцов изображений соответственно. Ясно, что s, d, p и v тоже являются изображениями с размерами $M \times N$. Заметим, что в так опреде-