



УДК 681.3

Фильтрация растровых изображений. Методические указания к лабораторной работе / Рязанская государственная радиотехническая академия; Сост.: И.А. Телков. Рязань, 2002. 15 с.

Содержит описание лабораторной работы, в которой рассматриваются простейшие алгоритмы фильтрации растровых изображений: свертка, медианная фильтрация, определение краев по Собелю.

Работа предназначена для использования в рамках лабораторного практикума по дисциплинам "Инженерная графика", "Компьютерная графика", "Компьютерная графика и геометрия" и "Компьютерная графика и геометрическое моделирование". Кроме того, справочные материалы лабораторной работы могут быть использованы в ходе курсового проектирования по дисциплинам "Машинная графика и геометрическое моделирование" и "Интерактивные графические системы", а также в ходе дипломного проектирования при создании пользовательского интерфейса.

Предназначена для студентов дневных и вечерних форм обучения специальностей 2203, 2204, 2205 и 0719.

Компьютерная графика, растровые данные, графические фильтры, свертка, медианная фильтрация.

Рецензент: кафедра САПР вычислительных средств Рязанской государственной радиотехнической академии.

Цель работы: изучение наиболее распространенных методов фильтрации растровой информации; освоение методов обработки растровой информации и эффективного программирования с учетом особенностей видеосистемы персонального компьютера.

1. Теоретическая часть

В данной лабораторной работе рассматривается класс преобразований растровых изображений, который называется **пространственными преобразованиями** (*spatial transforming*). Это термин появился гораздо раньше, чем на практике стали использоваться настоящие пространственные преобразования. В настоящее время широко используются трехмерные (пространственные) объекты и их преобразования. Использование подобного термина в данный момент может приводить к путанице. Поэтому в данной работе будем использовать более корректный термин - **двумерные**, или **2D-преобразования (2D-процессы)**. Аналогично пространственные фильтры (*spatial filters*) будем называть **2D-фильтрами**.

Подобные преобразования используют группы элементов изображения для формирования данных нового изображения. Этим 2D-процессы отличаются от процессов, которые используют информацию только одного элемента.

Группа элементов изображения, используемых в пространственных процессах, называется **областью примыкания**. Формально *область примыкания* - двумерная матрица значений элементов растрового изображения, каждое измерение которого имеет нечетное число элементов. Преобразуемая точка расположена в *центре области примыкания*. Учет точек в непосредственной близости от преобразуемой точки дает информацию о распределении яркости (цвета), которая используется большинством 2D-процессов.

Другим, более правильным термином для информации о распределении яркости является **пространственная частота**. **Пространственная частота** определяется как скорость изменения яркости элементов изображения или интенсивности, деленная на расстояние, на котором происходит это изменение. Пространственная частота имеет в изображении компоненты как горизонтального, так и вертикального направлений.

Изображение с высокой пространственной частотой содержит резкие близкорасположенные изменения в значениях соседних элементов. Изображение черно-белой шахматной доски является примером изображения с высокой пространственной частотой. Чем меньше квадраты этой доски, тем больше будет значение частоты.

Изображения с низкой пространственной частотой содержат большие области постоянных или медленно изменяющихся значений элементов изображения. Изображения облаков обычно содержат низкую пространственную частоту.

Информация о пространственной частоте позволяет использовать 2D-процессы в качестве фильтров путем удаления или усиления компонентов определенной частоты, обнаруженных в изображении. Таким образом, многие 2D-процессы попадают в категорию пространственных фильтров. Подобно электрическим фильтрам, пространственные фильтры описываются математическими формулами. Необходимо отметить, что сложные математические выкладки алгоритмов, используемые в процедурах фильтрации, имеют в значительной степени интуитивное объяснение. Пространственная фильтрация находит широкое применение в обработке изображений. Например, ее можно использовать для выделения некоторых признаков изображения (усиление границ и их определение), для увеличения резкости, сглаживания, размытия изображения и снятия случайных помех в изображении.

В рамках данной лабораторной работы рассматриваются следующие фильтры для обработки растровых изображений:

- свертка;
- медианная фильтрация;
- определение краев по Собелю.

При обработке изображений можно выделить следующие основные моменты при реализации алгоритмов:

- при фильтрации переход осуществляется через поэлементную обработку исходного изображения;
- каждый элемент входного изображения преобразуется в элемент нового изображения;
- новое значение элемента помещено в выходном буфере изображения в той же ячейке, которая была взята из входного буфера изображения;
- при масштабировании непрерывное изображение восстанавливается по заданному растровому изображению.

1.1. Алгоритмы, базирующиеся на свертке

Свертка - это алгоритм общего назначения, который можно использовать в различных преобразованиях растровых изображений (рис.2.1).

Свертка (*convolution*), или **свернутое матричное произведение**, представляет собой результат следующей матричной операции:

$$C = A \bullet B = \sum_i \sum_j a_{ij} \times b_{ij}.$$

Каждый элемент в *области примыкания* (на рис.2.1 она имеет размер 3x3) умножается на ядро свертки аналогичной размерности:

$$C = P \bullet K = \sum_i \sum_j p_{ij} \times k_{ij},$$

где p_{ij} - значения пикселей, из области примыкания, а k_{ij} - элементы ядра свертки, называемые **весовыми факторами**, или **коэффициентами свертки**.

Результат C при этом заменяет значение рассматриваемого элемента изображения. Размеры и структура весовых факторов, содержащихся в ядре свертки, определяют тип 2D-преобразования. Изменение весового фактора в пределах ядра свертки влияет на величину и, возможно, на знак общей суммы и, таким образом, воздействует на значение рассматриваемого элемента изображения. Большинство матриц ядер имеют размер 3x3 (или в общем случае - нечетное число рядов и колонок). Этот формат коэффициентов свертки в пределах ядра используется на практике в качестве стандарта. Увеличение размера ядра повышает гибкость процесса свертки. На рис. 2.1 представлена схема процесса свертки.

Область примыкания описывается матрицей $P = \|p_{ij}\|$, где p_{ij} - значения пикселей, входящих в область примыкания пикселя p_{yx} , расположенного в позиции x строки y . Для ядра размером 3x3:

$$i = (y-1) \div (y+1),$$
$$j = (x-1) \div (x+1).$$

При фильтрации изображений возникает ряд трудностей.

Первая и самая основная трудность связана с обработкой **краев изображения**. По мере того как мы перемещаем ядро свертки (и соответственно рассматриваемый элемент изображения) по изображению элемент за элементом, у нас возникнут сложности потому, что на границе изображения рассматриваемый элемент не имеет соседей со всех сторон. Другими словами, ядро свертки выходит краем за границы изображения. Для устранения такого явления можно использовать несколько методов. Наиболее простыми решениями являются:

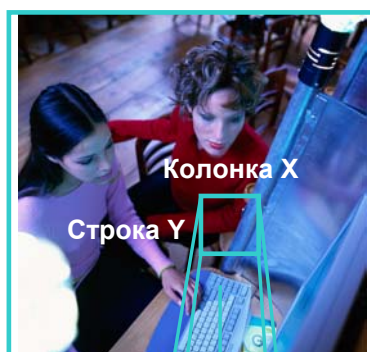
- 1) данные на краях изображения не учитываются или
- 2) данные изображения могут копироваться, для того чтобы "синтезировать" дополнительные данные границы.

Вторая трудность связана с **выходом новых значений** преобразуемых элементов **за пределы допустимых значений**. Для большинства ядер новые значения преобразуемых элементов находятся в пределах действительных значений от 0 до 255. К сожалению, ядро размытия дает значения, выпадающие из диапазона действительных значений элементов изображения. По этой причине в алгоритм свертки вводится масштабирование.

При внимательном рассмотрении становится понятным, почему ядро размытия дает такие большие значения, в отличие от других ядер. Во всех ядрах, за исключением ядра размытия, сумма всех весовых коэффициентов равна либо 0, либо 1. Сумма коэффициентов ядра размытия равна 25. Это означает, что расчет весовой суммы будет равен 25 значениям элементов изображения в области примыкания 5x5. Окончательный выбор параметра масштабирования является субъективным и зависит от вида изображения.

Также необходимо учесть в алгоритме свертки **знак значений элементов изображения**. Если ядро свертки содержит отрицательные вещественные коэффициенты, что характерно для большинства ядер, можно получить отрицательное значение интенсивности элементов изображения. По этой причине отрицательное значение элементов изображения при свертке устанавливается равное 0. Для того чтобы рассмотреть отрицательные значения интенсивности используются другие методы. Например, можно использовать абсолютное значение интенсивности вместо присваивания значения, равного 0. Другой метод состоит в масштабировании всех значений элементов таким образом, чтобы отрицательные значения стали равными 0.

Исходное изображение



$$\begin{matrix} P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 \\ P_7 & P_8 & P_9 \end{matrix}$$

Окружение точки 3*3

×

Ядро преобразования 3×3

$$\begin{matrix} K_1 & K_2 & K_3 \\ K_4 & K_5 & K_6 \\ K_7 & K_8 & K_9 \end{matrix}$$

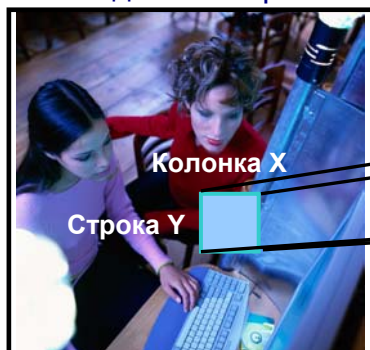
Рассчитывается значение P_5

Вычисление взвешенной суммы

$$\begin{aligned} & (K_1 * P_1) \\ & (K_2 * P_2) \\ & (K_3 * P_3) \\ & (K_4 * P_4) \\ & (K_5 * P_5) \\ & (K_6 * P_6) \\ & (K_7 * P_7) \\ & (K_8 * P_8) \\ & (K_9 * P_9) \end{aligned}$$

Новое значение P'_5

Выходное изображение



Новое значение P'_5 помещается в выходное изображение

Рис. 2.1. Реализация свертки

В рамках данной лабораторной работы предлагается реализовать следующие типы фильтров, использующие алгоритм свертки:

- низкочастотные пространственные фильтры;
- высокочастотные пространственные фильтры;
- усиление края методом сдвига и разности;
- усиление края методом выравнивающего фильтра;
- усиление края методом направленного градиента;
- усиление края по Лапласу;
- фильтры размытия;
- медианный фильтр.

Далее рассмотрим все вышеперечисленные типы фильтров, использующие алгоритм свертки.

1.1.1. Низкочастотные пространственные фильтры

Низкочастотные пространственные фильтры оставляют низкочастотные компоненты изображения нетронутыми и ослабляют высокочастотные компоненты.

Такие фильтры **используются для понижения визуального шума**, содержащегося в изображении, а также для удаления высокочастотных компонент из изображения с тем, чтобы можно было тщательнее исследовать содержание низкочастотных компонент. По мере того как уменьшается содержание высокочастотных компонент, могут быть идентифицированы более слабые изменения низкой частоты.

Частота отсечки низкочастотного фильтра определяется размером и коэффициентами ядра. Отметим, что сумма значений ядра для всех низкочастотных фильтров равна 1. Этот факт важен для понимания работы низкочастотных фильтров. Рассмотрим часть изображения, не содержащую высокочастотных компонент. Это означает, что все значения элементов изображения постоянные или они изменяются очень медленно. По мере того как низкочастотное ядро проходит через область изображения, новое значение преобразуемых элементов изображения (элементы, находящиеся вокруг ядра) вычисляется как сумма коэффициентов ядра, умноженных на значения элементов изображения в области примыкания. Если все значения элементов изображения в области примыкания одинаковы (постоянны), новые значения элементов изображения будут такими же, как и старые. По этой причине сумма коэффициентов выбрана равной 1. Сохраняется содержание низкочастотных компонент. По мере того как ядро движется по области изображения с содержанием высокочастотных компонент, любые быстрые изменения интенсивности усредняются с оставшимися элементами изображения в области примыкания, тем самым, понижая содержание высокочастотных компонент.

Визуальным результатом низкочастотной фильтрации является **слабая нерезкость изображения**. Она возникает потому, что любые увеличения значений элементов изображения усредняются с их окружением, что ослабляет содержание высокочастотных компонент.

Низкочастотная фильтрация может использоваться для улучшения резкости изображения. Если отфильтрованное изображение с помощью низкочастотного фильтра **вычитается** из исходного изображения, то в результате будет относительное увеличение содержания высокочастотных компонент без увеличения шума в изображении. Субъективно результирующее изображение кажется резче, чем исходное. Это может быть использовано для освещения областей изображения, которые затемнены туманом или облаками.

Примеры ядер низкочастотных фильтров:

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \quad \begin{pmatrix} 1/10 & 1/10 & 1/10 \\ 1/10 & 1/5 & 1/10 \\ 1/10 & 1/10 & 1/10 \end{pmatrix} \quad \begin{pmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{pmatrix}$$

На практике широко используется вариант размытия – **фильтр Гаусса**:

$$\begin{pmatrix} 0 & 1/5 & 0 \\ 1/5 & 1/5 & 1/5 \\ 0 & 1/5 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1/6 & 0 \\ 1/6 & 1/3 & 1/6 \\ 0 & 1/6 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1/8 & 0 \\ 1/8 & 1/2 & 1/8 \\ 0 & 1/8 & 0 \end{pmatrix}$$

1.1.2. Высокочастотные пространственные фильтры

Высокочастотные фильтры выделяют высокочастотные компоненты изображения, оставляя содержание низкочастотных компонент нетронутым.

Содержание низкочастотных компонент увеличивается по отношению к содержанию высокочастотных компонент. Такая фильтрация используется в тех случаях, когда необходимо исследовать объекты с высокой пространственной частотой. Области изображения с высокой частотой будут хорошо освещены (станут ярче), а части с низкой частотой станут черными. Иногда **резкость изображения увеличивается** после высокочастотной фильтрации за счет выделения шума изображения. При использовании высокочастотной фильтрации **возможно усиление края изображения**. Далее представлены три ядра высокочастотных фильтров. Большое значение коэффициента центра ядра определяет действие высокочастотного фильтра. По мере того как центр с большим коэффициентом перемещается по области изображения с высокой пространственной частотой (что означает большие изменения интенсивности элементов изображения), новое значение рассматриваемых элементов изображения многократно увеличивается. Меньшие отрицательные значения коэффициента ядра, сгруппированные вокруг центра, уменьшают эффект большого весового фактора. В конечном итоге, большие изменения интенсивности элементов изображения усиливаются, а области постоянной интенсивности элементов изображения останутся неизменными. Другими словами, области постоянной интенсивности элементов изображения (области низких пространственных частот) не подвергаются этим преобразованиям.

Ядра высокочастотных пространственных фильтров имеют следующий вид:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

1.1.3. Усиление края методом сдвига и разности

Другим процессом, который можно продемонстрировать, используя алгоритм свертки, является усиление края. Усиление края используется как *предварительный шаг в процессе извлечения признаков изображения*. Алгоритмы усиления края сжимают изображение до краев, оно уменьшается и, во многих случаях, полностью уничтожается. По этой причине *обработанное изображение может сильно отличаться от исходного изображения*. Яркость края после усиления пропорциональна изменению яркости, окружающей края в исходном изображении.

Хотя усиление края в основном используется в машинном зрении, оно, конечно, имеет и другие применения. Например, информация о крае, полученная в процессе его усиления, может быть использована в исходном изображении для усиления его резкости. Усиление края можно использовать как художественный метод для изготовления оригинальных изображений, которые могут затем ретушироваться в программах рисования для создания высокохудожественных изображений.

Алгоритм усиления края методом сдвига и разности усиливает края изображения, сдвигая изображение на элемент и затем вычитая сдвинутое изображение из исходного. Результатом вычитания является измерение наклона распределения яркости. В области постоянной интенсивности результатом вычитания будет наклон, равный нулю. Нулевой наклон - это значения черных элементов изображения. В области с большими изменениями интенсивности, например на крае, вычитание даст большое значение наклона, что приводит к появлению светлого элемента изображения. Чем больше разность интенсивностей, тем более «белым» будет результирующий элемент изображения. При работе с этим алгоритмом необходимо помнить, что могут появляться значения отрицательного наклона с переходом от белого к черному цвету. При этом следует применять абсолютное значение функции, чтобы алгоритм сдвига и разности мог регистрировать края как переходов от черного к белому, так и от белого к черному.

Когда используется это приближение для усиления вертикальных краев, изображение смещается влево на один элемент и затем вычитается из исходного. Для усиления горизонтальных краев изображение смещается вверх на один элемент и вычитается. Для усиления горизонтальных и вертикальных краев изображения перед вычитанием изображение сдвигается сначала влево на один элемент, а затем вверх.

Хотя это приближение кажется простым, его осуществление достаточно сложно. Поэтому вместо фактического сдвига изображения используется свертка для получения того же эффекта. Далее представлены ядра свертки, которые осуществляют усиление края методом сдвига и разности. Эти ядра визуально представляют алгоритм сдвига и разности. Рассмотрим, например, усиление вертикального края. Выше мы говорили, что для усиления вертикальных краев изображение сдвигается влево на один элемент и вычитается. С вертикальным ядром происходит то же самое: сравниваются два горизонтально примыкающих элемента изображения, чтобы найти наклон изменения яркости. Если этот процесс проходит по области с постоянной интенсивностью I , результат свертки будет равен нулю, потому что $-1 \cdot I + 1 \cdot I = 0$.

Если наблюдаются большие изменения интенсивности, результатом будет либо большое положительное число (для перехода от черного к белому), либо большое отрицательное число (для перехода от белого к черному). Интенсивность результирующего элемента будет прямо пропорциональна наклону изменения яркости.

Ядра фильтров для усиления края методом сдвига и разности имеют следующий вид (Вертикальные края, Горизонтальные края, Вертикальные и горизонтальные края):

$$\begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

1.1.4. Усиление края методом выравнивающего фильтра

Существует много специальных ядер свертки для усиления и определения краев изображения. Далее представлены два таких ядра, которые называются выравнивающими фильтрами потому, что они напоминают признаки краев, которые они усиливают. Это примеры больших ядер свертки, которые гарантируют более точное определение края за счет уменьшения производительности. Возможны еще более крупные ядра. Они содержат шаблоны или определения контуров предметов, подвергшихся исследованию в изображении. Если контур соответствует шаблону, края высвечиваются, и все другие части изображения становятся черными.

Ядра фильтров для усиления края методом выравнивающего фильтра имеют следующий вид (Вертикальные и Горизонтальные края):

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

1.1.5. Усиление края методом направленного градиента

Усиление края методом сдвига и разности позволяет усилить вертикальные и горизонтальные края в изображении. В реальных программах большинство алгоритмов усиления края, используя только ядро 3×3 , усиливают более одной полной вертикальной и горизонтальной линии. Ядро большего размера может быть использовано при увеличении требований к вертикальному и горизонтальному краям.

Иногда необходимо высветить и другие края изображения, не строго вертикальные или горизонтальные. Могут быть важны диагональные края. Избирательное высвечивание краев в различных направлениях используется иногда для того, чтобы компьютер «получил» общее представление об изображении. Метод направленного градиента как раз используется для этой цели. Для высвечивания краев в восьми различных направлениях используются 8 различных ядер свертки. Эти направления называются как стороны света: север, северо-восток, восток, юго-восток, юг, юго-запад, запад, северо-запад. Все эти ядра представлены ниже.

Если существует положительный наклон в направлении ядра, в выходном изображении будет помещен ярко раскрашенный элемент изображения. Интенсивность выходного элемента изображения будет зависеть от наклона изменения яркости. Чем больше наклон, тем ярче элемент. Например, градиент ядра «восток» будет усиливать край, который содержит переход от

черного к белому слева направо. Если сумма коэффициентов ядра равна нулю, то области постоянной яркости (низкочастотные пространственные компоненты) будут ослаблены, т. е. области постоянной яркости будут на выходе представлены черными элементами изображения.

Ядра фильтров для усиления края методом направленного градиента, имеют следующий вид (Север, Северо-восток, Восток, Юго-восток, Юг, Юго-запад, Запад, Северо-запад):

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{pmatrix}$$

1.1.6. Усиление края по Лапласу

Метод усиления края по Лапласу отличается от других методов тем, что он не зависит от направления, то есть края высвечиваются независимо от направления. Это преобразование является приближением лапласиана, известного в математике и электронике. Усиление края по Лапласу образует более резкий край, чем при других алгоритмах усиления. Дополнительно оно высвечивает края, имеющие как положительные, так и отрицательные изменения яркости. Поэтому усиление края по Лапласу находит применение во многих машинных процессах. Из математики, известно, что функция $f(x, y)$ Лапласа записывается в виде:

$$L(f(x,y)) = d^2f/dx^2 + d^2f/dy^2$$

где d^2f/dx^2 - вторая частная производная по x , а

d^2f/dy^2 - вторая частная производная по y .

Для дискретных функций вторые производные могут быть аппроксимированы следующим образом:

$$d^2f/dx^2 = f(x+1) - 2*f(x) + f(x-1)$$

$$d^2f/dy^2 = f(y+1) - 2*f(y) + f(y-1)$$

Таким образом, лапласиан можно записать в следующем виде:

$$L(f(x,y)) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4*f(x,y)$$

Это выражение можно рассматривать как ядро свертки, записанное в виде функции $f(x, y)$. Само ядро при этом имеет вид:

$$P(x,y) = \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

Далее это ядро обозначено как LAP1. Все операции по усилению ядра, включая лапласианы, ослабляют низкочастотные пространственные компоненты изображения. Области постоянной интенсивности или линейно возрастающей интенсивности становятся черными как результат этих преобразований, а области быстро изменяющихся значений интенсивности ярко высвечиваются. Ядра свертки, ослабляющие низкочастотные компоненты, имеют коэффициенты, сумма которых равна нулю.

При применении ядер свертки типа лапласиан к различным изображениям можно получить некоторые интересные эффекты. Такой эффект достигается применением ядра LAP3. Это имеет эффект прямого добавления свернутых изображений обратно к исходному изображению, которое прибавляет неожиданные детали к неинтересному изображению.

Ядра лапласианов могут иметь следующий вид:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

1.1.7 Фильтры размытия

Может показаться, что в обработке изображений размытие изображения умышленно противопоставляется идее усиления изображения. Действительно, размытие изображения практически не находит применений в промышленности, хотя этот метод иногда применяется. Например, как художественный прием, размытие иногда может быть использовано для того, чтобы создать размытый задний план, на фоне которого помещены предметы переднего плана. Контраст резкости и размытия может быть приятен для глаза. Для этой цели используются фотографические фильтры. Размытие изображения можно получить, используя свертку. Далее представлены размеры ядер, используемые для размытия изображения. Это ядро, например, размером 3x3, содержащее единичные элементы. В сущности свертка с ядром размытия усредняет все значения элементов в области примыкания. Усреднение вызывает уменьшение деталей изображения, следствием которого является эффект размытия.

В работе предлагается реализовать следующие размеры ядер: 3x3, 5x5, 7x7, 11x11, 13x13, 15x15.

В качестве примера ядра можно привести следующее матрицы:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

1.1.8. Фильтр рельефа

Фильтр рельефа является разновидностью фильтра выделения краев. Рельефные фильтры преобразуют исходное цветное изображение в ахроматическое (grayscale) с объемным эффектом. Простейший вариант ядра фильтра имеет размер 3x3 и нулевую сумму элементов матрицы фильтра. Матрица ядра содержит только два ненулевых элемента: +1 и -1, которые расположены симметрично относительно центрального нулевого элемента матрицы. Работу рельефного фильтра можно рассматривать как вычитание изображения, смещенного относительно оригинала. Позиции ненулевых элементов матрицы ядра фильтра определяют направление рельефного выделения. Для усиления или уменьшения эффекта рельефа можно использовать масштабный коэффициент.

Примеры ядер рельефного фильтра:

$$\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

1.2. Медианная фильтрация

1.2.1. Алгоритм усредняющего фильтра

Усредненное фильтрование - пространственный процесс, который не попадает под категорию свертки.

Усредненное фильтрование использует значение элементов, содержащихся в области примыкания, для определения нового значения рассматриваемого элемента. Алгоритм позволяет работать со следующими размерами области примыкания: 3x3, 5x5, 7x7, 9x9, 11x11, 13x13, 15x15 и т.д. Однако новое значение элемента вычисляется алгоритмически после расположения элементов области примыкания в возрастающем порядке. В качестве нового значения выбирается среднее значение.

Результатом усредненного фильтрования является то, что любой случайный шум, содержащийся в изображении, будет эффективно устранен. Это происходит потому, что любое случайное резкое изменение в интенсивности элемента в пределах области примыкания, будет сортироваться, т. е. оно будет помещено либо на вершину, либо на нижнюю часть отсортированных значений области примыкания и не будет учитываться, так как для нового значения элемента всегда отбирается усредненное значение. Отбор случайных значений элементов показан на схеме. Применение усредненного фильтрования можно увидеть в визуальных эффектах.

Далее на схеме показан алгоритм усредняющего фильтра:

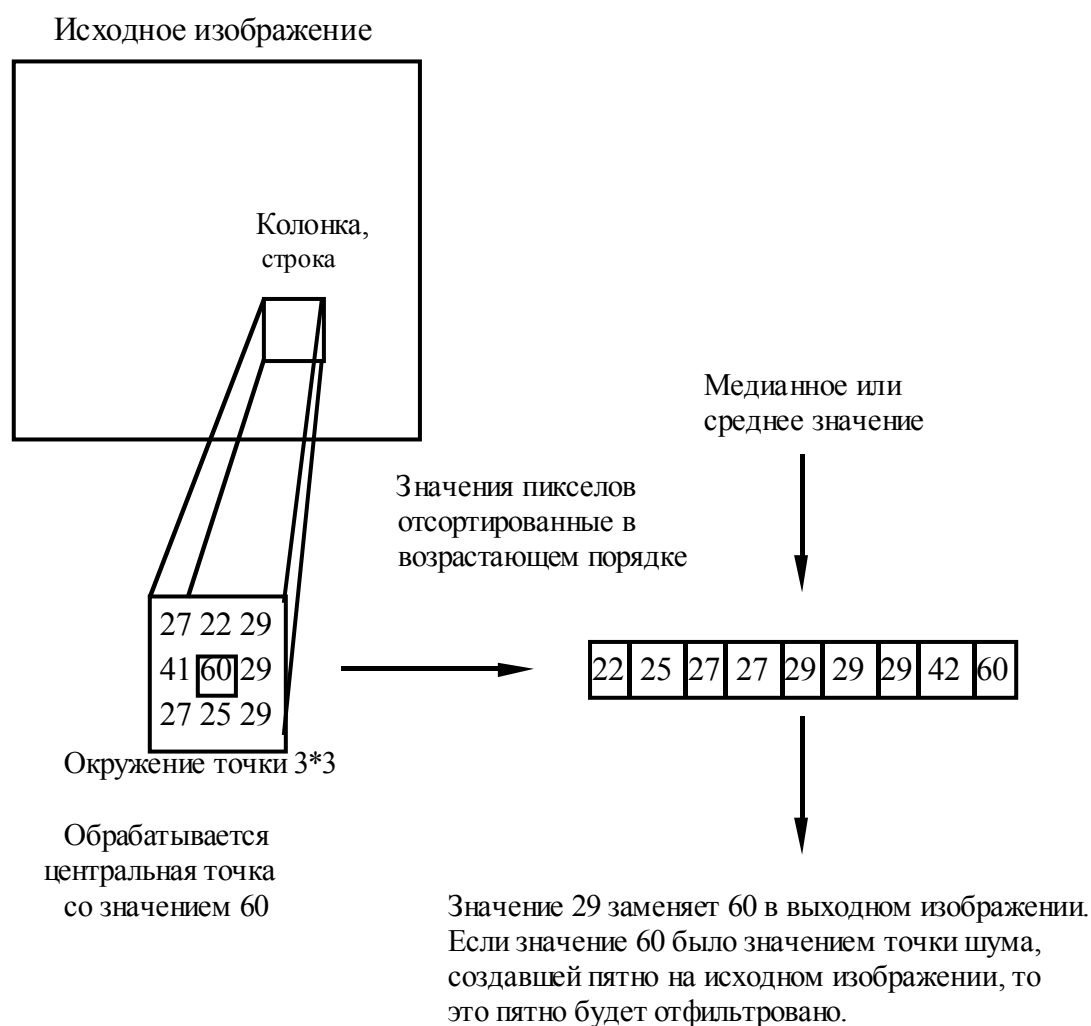


Рис. 2.2. Медианная фильтрация

1.2.2. Модификация алгоритма усредняющего фильтра

В данной библиотеке реализован алгоритм усредняющего фильтра, несколько отличающийся от классического. Отличие заключается во вводе дополнительного параметра - порога обработки. Смысл его раскрывается ниже.

При нахождении среднего элемента точки окружения вычисляется модуль разности между обрабатываемым элементом и средним элементом. Данная разность может находиться в диапазоне от 0 до 255. Если указанный модуль разности превышает порог обработки, то результирующий цвет заменяется на среднее значение точки окружения, иначе – нет.

При данной модификации алгоритма производится меньшая потеря уровня полезного сигнала, но в тоже время при правильном подборе порога обработки осуществляется почти одинаковое гашение шумов по сравнению с не модифицированным алгоритмом.

1.3. Определение краев по Собелю

Алгоритм Собеля - единственный нелинейный метод определения края. Этот метод нашел широкое применение. Это другой пример алгоритма со сложной математической базой, которая относительно проста для выполнения. Фактически имеются два различных метода выполнения алгоритма Собеля. В первом методе вычисляются 2 различные свертки:

ядро X			ядро Y		
-1	0	1	1	2	1
-2	0	-2	0	0	0
-1	0	1	-1	-2	-1

и, исходя из этих свертки, вычисляется величина и направление краев:

- величина = Квадратный корень ($X^2 + Y^2$)
- направление = $\arctg(Y/X)$

Это слишком сложный процесс, чтобы выполнять его для каждого элемента изображения. По этой причине был предложен другой метод выполнения алгоритма Собеля. Для ускорения большого числа вычислений используются различные сокращения. Метод, который будет использоваться, очень прост. Во-первых, предположим, что область примыкания точек 3x3 обозначена следующим образом:

а б в
г д е
ж з и

Через эту область могут быть проведены только четыре линии, чтобы пересечь среднюю точку (д). Эти линии следующие:

- линия 1: а – д – и
- линия 2: б – д – з
- линия 3: в – д – ж
- линия 4: е – д – г

Каждая линия, прочерченная через область примыкания, делит пространство на две трехточечные области. Например, линия 1 делит пространство на области (г, ж, з) и (б, в, е).

Для каждой из четырех линий вычисляется абсолютное значение разности средних величин двух подобластей. Таким образом, выполняется четыре расчета полных разностей. Рассматриваемому элементу изображения присваивается самое большое значение из четырех абсолютных разностей.

После применения алгоритма Собеля к каждому элементу изображения выходное изображение обычно подвергается обработке пороговым точечным процессом. Если новое значение рассматриваемого элемента (только что вычисленная самая большая разность) достигает или превышает определенный порог, цвет выходного элемента становится белым (если была произведена обработка компонента определенного цвета, то этой компоненте присваивается значение равное 255). Если значение меньше порога, элемент становится черным (если была произведена обработка компонента определенного цвета, то этой компоненте присваивается значение равное нулю). Чистым результатом применения алгоритма Собеля, следующим за точечным пороговым процессом, является черное или белое изображение, не содержащее информации об исходном изображении, исключая информацию о крае, но это не относится к случаю обработки конкретной компоненты цвета (красной, зеленой или синей), когда изменяется значение только выбранной компоненты и, следовательно, информацию о крае будет содержать только выбранная компонента.

2. НАПИСАНИЕ ФИЛЬТРОВ

При написании **программы фильтрации** необходимо реализовать следующие процедуры:

- открытие файлов формата *JPEG* и *BMP*;
- фильтрация изображения с использованием заданного фильтра;
- сохранение файлов формата *JPEG* и *BMP*.

Процедуры открытия и сохранения файлов реализуются с использованием стандартных методов языка *Object Pascal*, поэтому основная часть работы реализуется в процедуре/функции фильтрации изображения.

Вычисление нового значения пикселя $p(y,x)$ с использованием операции свертки для области примыкания 3×3 осуществляется по формуле:

$$p'(y,x) = C = \sum_{i=-1}^1 \sum_{j=-1}^1 p(y-2+i, x-2+j) \times k_{ij}.$$

где $p(i,j)$ – исходные значения пикселей, $p'(y,x)$ – новые значения пикселей, k_{ij} – значения элементов матрицы ядра свертки.

Для обращения к отдельным пикселям объекта *TBitmap* можно воспользоваться свойством *Pixels[x,y]* объекта *Canvas*. Однако это свойство является скрытым обращением к *Windows GDI* и имеет достаточно большое время реализации, что сказывается на быстродействии процедур фильтрации. Поэтому с целью повешения быстродействия рекомендуется использовать свойство *ScanLine[y]* объекта *TBitmap*.

Кроме того, следует помнить, что объекты класса *TJPEGImage* не имеют свойства *TBitmap*, что не позволяет напрямую обращаться к матрице пикселей. Поэтому перед фильтрацией файлов формата *JPEG* их необходимо преобразовать в объекты класса *TBitmap*. Следующий фрагмент кода, написанный для *Delphi*, использует для обращения к файлу потока и метод *Assign* для преобразования данных объекта *TJPEGImage* в данные объекта *TBitmap*:

```
var
  f: TFileStream;
  ji: TJPEGImage;
  bi: TBitmap;
...
begin
  f := TFileStream.Create(FileName, fmOpenRead);
  ji := TJPEGImage.Create;
  ji.LoadFromStream(f);
  bi := TBitmap.Create;
  bi.Assign(ji);
  ji.Free;
  Image1.Picture.Bitmap := bi;
  bi.Free;
...
end;
```

При написании процедуры (функции) фильтрации необходимо учитывать, что новые значения пикселя могут выходить за пределы диапазона допустимых значений. Для устранения этого недостатка следует учитывать следующие моменты:

- 1) весовой коэффициент свертки;
- 2) характер диапазона получаемых значений;
- 3) границы диапазона получаемых значений.

Весовой коэффициент L свертки K зависит от значения суммы элементов $S(K)$ матрицы ядра свертки $K = \| k_{ij} \|$:

$$S(K) = \sum_i \sum_j k_{ij}.$$

Для большинства фильтров она составляет либо 0, либо 1. Только для фильтра размытия она равна $n \times n$, где n – размер ядра свертки. Новое значение пикселя в этом случае вычисляется по следующей формуле:

$$p''(y,x) = p'(y,x)/L,$$

где величина L определяется выражением:

$$L = \begin{cases} 1, & \text{если } S(K) = 1 \text{ или } S(K) = 0; \\ S(K), & \text{если } S(K) > 1. \end{cases}$$

Характер диапазона. В том случае, если величина $S(K) = 0$, то центр диапазона получаемых значений совпадает с нулем, а сами величины имеют как положительные, так и отрицательные значения. Поэтому для приведения их в стандартный диапазон значений пикселей $[0; 255]$ необходимо произвести смещение полученных значений на величину половины диапазона $H=128$:

$$p''(y,x) = p'(y,x) + 128.$$

Интенсивность результата. Интенсивность результирующего изображения W требует анализа изображения и определяется достаточно сложно. Поэтому ее рекомендуется определить как один из параметров настройки фильтра (по умолчанию можно устанавливать значение $W=1$).

Таким образом, итоговая формула для вычисления нового значения пикселя после выполнения операции свертки имеет вид:

$$p''(y,x) = W \times p'(y,x) / L + H,$$

где W – коэффициент интенсивности, $p'(y,x)$ – результат свертки, L – весовой коэффициент свертки и H – смещение диапазона значений.

Границы диапазона. После выполнения перечисленных операций достаточно часто получаемый результат все равно выходит за пределы диапазона допустимых значений величин пикселей. В простейшем случае (рекомендуемом в данной лабораторной работе) эта проблема решается путем простого отсечения получаемых величин. Например, для языка Паскаль:

```
If p<0 then p:=0;  
If p>255 then p:=255;
```

или для языка Си:

```
if (p<0) p=0;  
if (p>255) p=255;
```

Фильтрация полноцветных изображений. Все описанное выше относилось к однослойным ахроматическим изображениям (англ. - *grayscale*), цветовая глубина которых составляет 1 байт/пиксель. При обработке полноцветных изображений в формате *TrueColor RGB888* (3 байта/пиксель – по байту на канал красного, зеленого и синего цвета) необходимо обрабатывать каждый цветовой канал отдельно. Формулы фильтрации для каждого цветового канала остаются теми же, что и для ахроматических изображений.

3. ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ РАБОТЫ

1. **Ознакомиться с теоретической частью** лабораторной работы. Получить у преподавателя задание на разработку программы фильтрации растровых изображений и тестовые файлы.
2. **Создать программу**, позволяющую загружать растровое изображение в формате *Windows Bitmap* (или в формате *JPEG*) в отдельное окно.
3. Сформировать исходные данные, необходимые для выполнения условия для проверки корректности растрового файла. Включить их в разрабатываемую программу и **предъявить ее преподавателю**.
4. Реализовать алгоритм, соответствующий варианту задания, и опробовать его на различных растровых файлах. Рекомендуется первоначально осуществлять отладку на ахроматических файлах (их имена имеют префикс «gs» - *grayscale*), а затем перейти к полноцветным изображениям. **Предъявить результаты работы программы преподавателю**

Варианты заданий

№	Фильтр	Ядро фильтра	Раздел
1	НЧФ – Сглаживание	3x3	1.1.1
2	Размытие (равномерное)	3x3, 5x5, 7x7, 9x9 и т.д.	1.1.7
3	Усиление края методом градиента	3x3	1.1.5
4	Гауссово размытие	3x3	1.1.1
5	ВЧФ – Резкость	3x3	1.1.2
6	Усиление края по Лапласу	3x3	1.1.6
7	Усиление края методом сдвига и разности	3x3	1.1.3
8	Фильтр рельефа	3x3	1.1.8
9	НЧФ - Размытие (с центр. элементом)	3x3, 5x5, 7x7, 9x9 и т.д.	1.1.1
10	ВЧФ – Резкость	3x3, 5x5, 7x7, 9x9 и т.д.	1.1.2
11	Медианная фильтрация	3x3	1.2
12	Усиление края методом выравнивающего фильтра	3x5, 5x3 и т.д.	1.1.4

4. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Линдли К. Практическая обработка изображений на языке Си. М.: Мир, 1996.
2. Шлихт Г.Ю. Цифровая обработка цветных изображений. М.: ЭКОМ, 1997.
3. Юань Фень. Программирование графики для Windows. СПб.: Питер, 2002.