

ЛАБОРАТОРНАЯ РАБОТА 1. ЦИФРОВАЯ ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ

1.1. Цифровые фильтры

Рассмотрим дискретное описание линейных преобразований (операторов), которые реализуют программно или с помощью *цифровых процессоров*, например, на ПЛИС. Обработке подлежат цифровые сигналы, которые получены после дискретизации и квантования. Цифровую фильтрацию широко используют при обработке изображений. Принцип цифровой фильтрации состоит в том, что значение элемента изображения модифицируют с учетом значений соседних элементов. Цифровые фильтры применяют для увеличения *резкости* изображения (двумерная апертурная коррекция) или *сглаживания* изображения с целью уменьшения заметности шумов; гребенчатые фильтры используют для разделения спектров сигналов яркости и цветности в системах *NTSC* и *PAL*. Другое применение цифровой фильтрации –препарирование изображений: выделения характерных особенностей изображений, например, *контуров*, *локальных экстремумов яркости*; и т.д. Цифровые фильтры могут быть *линейными*, *нелинейными* и *логическими*.

Линейные преобразования характерны тем, что для них справедлив принцип суперпозиции. Цифровой фильтр считают линейным, если: сумма любого числа входных сигналов преобразуется в сумму откликов на отдельные слагаемые. Пусть последовательности $a_1(k)$ и $a_2(k)$ преобразованы с помощью цифрового фильтра F в $b_1(k) = F[a_1(k)]$ и $b_2(k) = F[a_2(k)]$, соответственно, а c_1 и c_2 постоянные коэффициенты. Тогда справедливо: $b(k) = F[c_1 a_1(k) + c_2 a_2(k)] = c_1 b_1(k) + c_2 b_2(k)$.

При сдвиге входной последовательности на m отсчетов, где m - целое число, выходная последовательность также сдвигается на m : если $y(k) = F[x(k)]$, то $F[x(k - m)] = y(k - m)$.

Фильтр называют стационарным, если его импульсная характеристика не зависит от времени. В общем виде сигнал - вектор, а оператор, который

преобразует один вектор в другой - матрица. Частным видом линейного оператора (линейного цифрового фильтра) являются операторы инвариантные к сдвигу: их свойства не меняются при перемещении относительно сигнала. Импульсная реакция такого фильтра есть функция разности аргументов $h(t, \tau) = h(t - \tau)$, так что применительно к аналоговому сигналу

$$b(t) = \int_{-\infty}^{\infty} a(\tau)h(t - \tau)d\tau, \text{ причем, эту операцию называют сверткой (рис. 1.1).}$$

Преобразование Фурье от импульсной реакции линейного оператора (фильтра) инвариантного к сдвигу является его *частотной характеристикой*:

$$H(f) = \int_{-\infty}^{\infty} h(t)\exp(-j2\pi ft)dt.$$

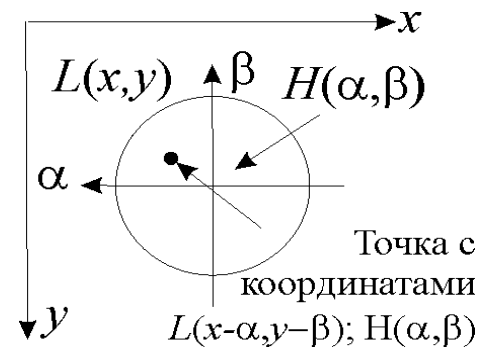


Рис. 1.1. Двумерная свертка двух функций

Преобразование Фурье от свертки двух функций равно произведению спектров этих функций: $\Phi\{L(x) * H(x)\} = F(f)H(f)$. Это основа двойственности фильтрации. В сигнальной плоскости фильтрацию проводят путем свертки сигнала с импульсной реакцией фильтра; в частотной - путем перемножения спектра сигнала и частотной характеристики фильтра.

Достоинства цифровых фильтров, по сравнению с аналоговыми фильтрами, реализованными на индуктивности, емкости и сопротивлении, несомненны:

- абсолютная стабильность характеристик, не зависящая от воздействия внешних условий;
- простота изменения характеристик, поэтому такие фильтры можно использовать в качестве адаптивных устройств (с перестраиваемыми характеристиками);

- при эксплуатации не возникает проблемы согласования нагрузок;
- могут работать в диапазоне сверхнизких частот;
- имеют линейные фазовые характеристики и т. д.

Недостатком цифровых фильтров является работа с цифровыми сигналами, что неизбежно приводит к некоторым ошибкам.

Процесс фильтрации заключается в том, что отсчеты фильтрованного сигнала $\{b_k\}$ выражают через отсчеты исходного сигнала $\{a_k\}$ и отсчеты усеченной по спектру (чтобы ее можно было выразить через отдельные отсчеты по Котельникову) импульсной реакции фильтра $\{h_k\}$. Цифровой фильтр преобразует последовательность отсчетов входного сигнала a_1, a_2, \dots, a_n в выходную последовательность b_1, b_2, \dots, b_n , причем выходной сигнал в каждый момент времени представляет собой взвешенную сумму входных

$$\text{сигналов: } b_k = \sum_{n=-\infty}^{\infty} a_n h_{k-n} = \sum_{n=-\infty}^{\infty} a_{k-n} h_n ,$$

Пользоваться такой формулой невозможно, так как бесконечные пределы физически не реализуемы. Поэтому используют конечномерное приближение к непрерывному фильтру, отбрасывая члены с достаточно малыми значениями h_n , в результате чего получают некоторую маску (в данном случае одномерную), размером $2N + 1$ отсчетов (число элементов в маске чаще всего нечетное, чтобы фильтр был симметричным)

$$b_k \approx a_{k-N} h_N + \dots + a_k h_0 + \dots + a_{k+N} h_{-N} = \sum_{n=-N}^N a_{k-n} h_n \quad (1.1)$$

Это алгоритм *не рекурсивного*, или *трансверсального* фильтра (рис. 1.2). Выходная переменная зависит только от $(2N + 1)$ значений входной переменной. Это фильтры с *конечной памятью* (КИХ-фильтры).

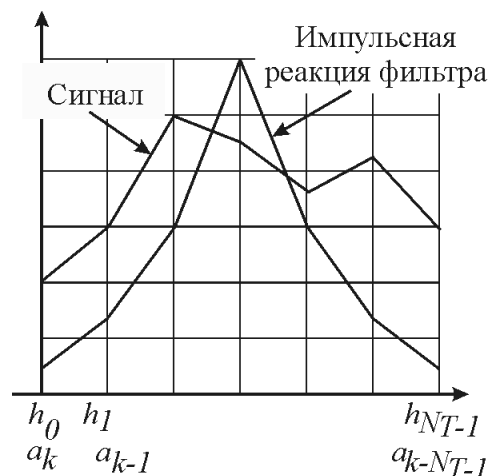


Рис. 1.2. Другое обозначение отсчетов сигнала и импульсной реакции

Цифровой фильтр, реализованный непосредственно по выражению (1.1), для вычисления одного отсчета преобразованного сигнала требует выполнения $2N + 1$ операций умножения и $2N$ операций сложения. Сократить объем вычислений можно путем использования *рекурсивных цифровых фильтров*, которые имеют более сложную структуру, а выходной сигнал является функцией предшествующих значений как входного, так и выходного сигналов:

$$b_k = h_0 a_k + h_1 a_{k-1} + \dots + h_{N_T-1} a_{k-N_T+1} + g_1 b_{k-1} + g_2 b_{k-2} + \dots + g_{N_P} b_{k-N_P};$$

$$b_k = \sum_{n=0}^{N_T-1} h_n a_{k-n} + \sum_{n=1}^{N_P} g_n b_{k-n}. \quad (1.2)$$

В этом случае по-другому обозначены отсчеты импульсной реакции трансверсальной части фильтра (рис. 1.3). В результате фильтр вносит чистую задержку (выходной сигнал сдвинут на $N_T/2$). Рекурсивные фильтры позволяют получить заданную амплитудно-частотную характеристику АЧХ при меньшем числе звеньев и с более точным соответствием аналоговому прототипу. Фильтры относятся к системам с обратной связью и нуждаются в анализе устойчивости. Поскольку выходная переменная у таких фильтров в данный момент времени зависит не только от значений входных переменных, но и от значений выходной переменной в предыдущие моменты времени, это фильтры с *бесконечной памятью* (БИХ-фильтры).

Вычисление последовательности значений $\{b_k\}$ для рекурсивных фильтров требует меньшего числа операций на один отсчет, так как *использованы результаты предыдущих вычислений*. Это преимущество в быстродействии заставляет искать возможности аппроксимации требуемого фильтра рекурсивным фильтром.

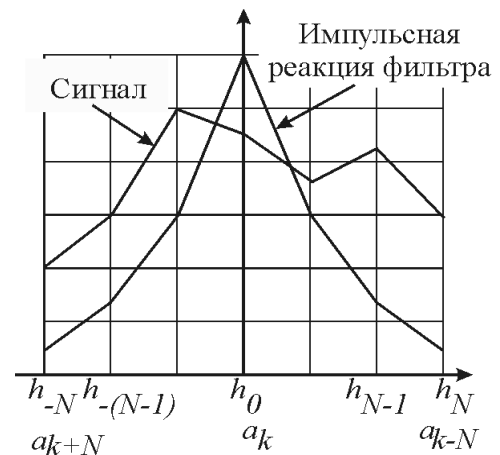


Рис. 1.3. Дискретная свертка сигнала и импульсной реакции фильтра

Рассмотрим часто применяемый при обработке изображений фильтр, вычисляющий текущее среднее значений сигнала на данном отрезке (окне)

$$b_k = \frac{1}{2N+1} \sum_{n=-N}^N a_{k-n} \quad (1.3)$$

где отсчеты импульсной реакции фильтра

$$h_n = \frac{1}{2N+1}, \text{ где } n = -N \dots 0 \dots N.$$

Выражение (1.3) можно преобразовать в рекурсивную форму

$$b_k = (a_{k+N} - a_{k-N-1}) / (2N+1) + b_{k-1} \quad (1.4)$$

Построив вычисления по выражению (1.4) можно найти текущее среднее не за $2N$ сложений на один отсчет среднего, а только за три. Замечательно то, что число операций в этом случае не зависит от числа отсчетов, по которым происходит усреднение. Если задать апертуру (маску) фильтра в три элемента $n = \{-1, 0, +1\}$, то не очень видны преимущества рекурсивного вычисления. Для фильтра из пяти элементов $n = \{-2, -1, 0, +1, 2\}$ по выражению (1.3) нужно сделать 5 сложений и одно деление, а по выражению (1.4) - по-прежнему 3 сложения и одно деление. Выигрыш тем значительнее, чем больше размер апертуры (рис. 1.4).

При обработке изображений используют двумерные линейные преобразования. Для двумерного случая выражение (1.1) можно переписать в следующем виде:

$$b_{kl} \approx \sum_{m=-N}^N \sum_{n=-N}^N a_{l-m, k-n} h_{m,n}, \quad (1.5)$$

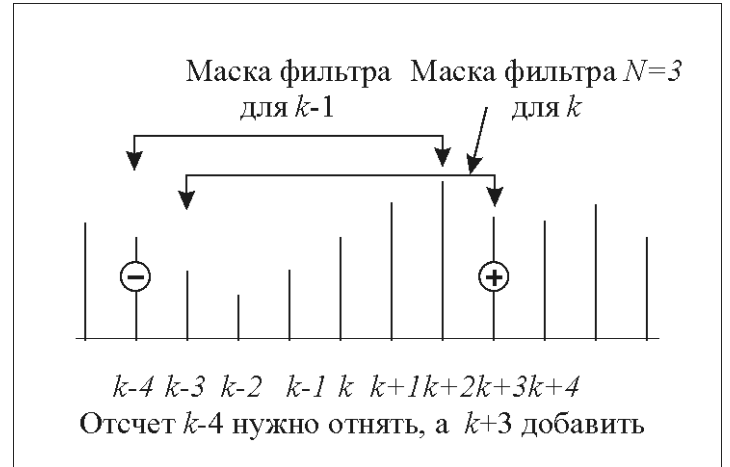


Рис. 1.4. Схема вычисления текущего среднего

где $\{b_{lk}\}; \{a_{lk}\}; \{h_{mn}\}$ взятые на прямоугольном окне отсчеты сигналов и импульсной реакции фильтра. Это математическая запись алгоритма, который реализуют все программы фильтрации изображений.

В двумерном случае также возможно построение рекурсивных фильтров, но при этом необходимо задаться направлением рекурсии. В соответствии с телевизионной разверткой логично принять «прошлыми» значения сигнала сверху и слева от данного отсчета и нумеровать отсчеты слева направо и сверху вниз. Тогда формулу для одномерного рекурсивного фильтра можно обобщить на двумерный случай следующим образом:

$$b_{kl} = \sum_{m=0}^{N_T-1} \sum_{n=0}^{N_T-1} a_{l-m, k-n} h_{mn} + \sum_{m=1}^{N_P} \sum_{n=-N_P}^{N_P} b_{l-m, k-n} g_{m,n} + \sum_{n=1}^{N_P} b_{l, k-n} g_{0,n}. \quad (1.6)$$

Существует еще один класс двумерных цифровых фильтров, представляющих особый интерес с точки зрения экономии вычислительных затрат – это *двумерные разделимые фильтры*. Импульсная реакция для таких фильтров может быть представлена в виде произведения одномерных функций $h_{m,n} = h_m h_n$. Тогда формула (1.6) принимает вид

$$b_{lk} = \sum_{m=-N}^N h_m \sum_{n=-N}^N h_n a_{l-m, k-n},$$

и вычисления можно производить рекурсивно. Примером является фильтр, используемый для получения текущего среднего значения сигнала по квадратному окну $N * N$

$$b_{lk} = \frac{1}{(2N+1)^2} \sum_{m=-N}^N \sum_{n=-N}^N a_{l-m, k-n}.$$

1.2. Синтез цифровых фильтров

Чтобы синтезировать цифровой фильтр нужно задать его частотную характеристику и затем найти отсчеты импульсной реакции h_n . Можно действовать в обратном порядке: задать импульсную реакцию, а затем

рассчитать частотную характеристику. Частотная характеристика и импульсная реакция фильтра связаны через преобразование Фурье.

Различают два класса цифровых фильтров: *фильтры нижних частот* и *фильтры верхних частот*.

Фильтры нижних частот применяют для сглаживания

изображений, снижения влияния зернистости изображений из-за шумов, подавления помех дискретизации и ослабления муар эффекта. Простым примером является сглаживание изображений путем вычисления средних значений элементов изображения в некоторой области. Соответствующую функцию содержат многие программы обработки изображений. Для нее задают число усредняемых элементов по координатам x и y . Коэффициенты, на которые при фильтрации умножают значения сигнала, сводят в матрицу \mathbf{F} (*матрицу свертки, маску или апертуру*) размером $n \times n$. При фильтрации апертуру перемещают по битовой матрице как маску, которая определяет, какие элементы изображения нужно обрабатывать в данный момент времени и на какие коэффициенты их следует умножать (рис. 1.5). Например, при усреднении по трем элементам по горизонтали и вертикали матрица \mathbf{F} размером 3×3 состоит из единиц:

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} / k. \quad (1.7)$$

Элементы, которые охватывает маска, умножают на соответствующие им коэффициенты, результаты складывают и делят на нормирующее число k , равное сумме коэффициентов маски (в данном случае $k=9$). Операцию деления

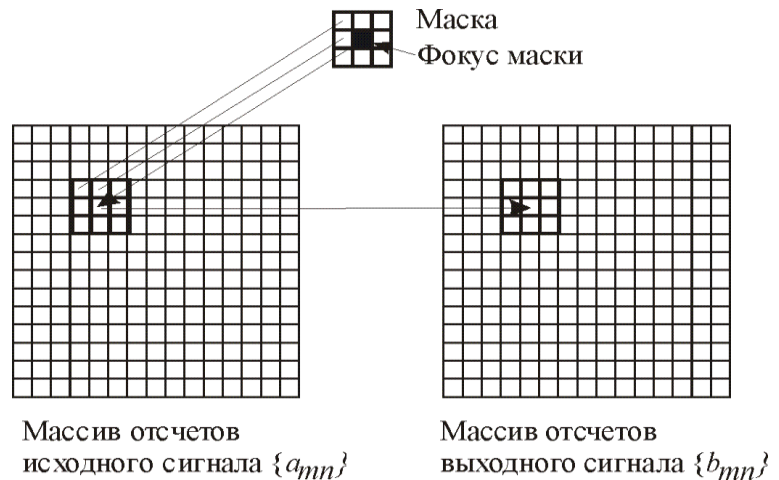


Рис. 1.5. Двумерная фильтрация

можно исключить, если сумма коэффициентов в маске всегда будет равна единице.

Фокусом маски называют ее центральный элемент (см. рис. 1.5). Все операции производят с исходной битовой матрицей, а в выходной матрице полученное значение элемента ставят на место, где в настоящее время находится центральный элемент маски. Затем маску передвигают на один элемент. Исключение составляет операция “Мозаика”, когда всем попадающим под маску элементам присваивают одно и то же вычисленное значение, а маску сдвигают на n элементов.

Значения коэффициентов в матрице \mathbf{F} представляют собой отсчеты *двумерного импульсного отклика* изображающей системы на воздействие также *двумерной дельта функции*. Для оптической системы – это *функция рассеяния точки*; для телевизионной – *апертура*. Параметрами апертуры являются ее диаметр, форма и закон распределения прозрачности. При цифровой фильтрации диаметр апертуры – это размерность матрицы маски, значения ее коэффициентов – отсчеты функции распределения прозрачности $\rho(\alpha, \beta)$, а форму апертуры обычно принимают квадратной. Например, матрица \mathbf{F} в выражении (1.7) соответствует квадратной апертуре с равномерным законом распределения прозрачности. Увеличение диаметра апертуры достигают повышением размерности матрицы \mathbf{F} , при этом сглаживающее действие цифрового фильтра возрастет, а закон распределения прозрачности может быть воспроизведен точнее за счет большего числа отсчетов. При сглаживании изображений вычисляют *взвешенное среднее* соседних элементов изображения. Действие типичной маски для сглаживания изображения

$$\mathbf{F} = \begin{bmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{bmatrix} / k$$

с нормирующим числом $k = 36$ иллюстрирует изображение на рис. 1.6.

Фильтры верхних частот применяют для повышения резкости изображения. Резкость изображения можно повысить, если усилить скачки яркости на границах объекта. Это так называемая *апертурная коррекция*. Сглаживающее действие апертуры можно скомпенсировать, если из сигнала U_{ij} , соответствующего фокусу апертуры (см. рис. 1.5), вычесть сигналы соседних элементов. Тогда матрица коэффициентов для фильтрации примет следующий вид:

$$\mathbf{F} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & X & -1 \\ -1 & -1 & -1 \end{bmatrix} / k ,$$

где коэффициент X определяет глубину коррекции, а нормирующий множитель k равен сумме коэффициентов маски.



Рис. 1.6. Результат действия фильтра НЧ

Для матрицы \mathbf{F} размером 3×3 коэффициент X вычисляют по выражению:

$$X = \text{int}[(100/S - 1) + 8],$$

где S – глубина коррекции в процентах, функция int означает округление до следующего целого числа.

Применять апертурную коррекцию надо с осторожностью, так как побочным эффектом является увеличение шума в изображении. Коэффициент S редко назначают большим, чем $(20 \div 30)\%$. Если соотношение сигнал-шум в исходном изображении недостаточно высокое, апертурную коррекцию применять нельзя, а для субъективного улучшения качества изображения лучше применить сглаживающий фильтр.

Частотной характеристикой цифрового фильтра, определяемого соотношением (1.1), называется частотная характеристика соответствующего ему непрерывного фильтра с импульсной реакцией

$$h(t) = \sum_{n=-N}^N h_n \delta(t - \frac{n}{2F}),$$

здесь $2F$ - частота дискретизации и F - верхняя граничная частота исходного сигнала и частотной характеристики фильтра, а обратная ей величина $1/2F$ – это шаг дискретизации, причем применительно к битовой матрице – это расстояние между пикселями. Тогда

$$H(f) = \int_{-\infty}^{\infty} \left[\sum_{n=-N}^N h_n \delta(t - \frac{n}{2F}) \right] \exp(j2\pi ft) dt = \sum_{n=-N}^N h_n \exp(j2\pi f \frac{n}{2F}). \quad (1.8)$$

Это выражение получилось в силу свойств дельта функции при ее интегрировании. Найдем связь отсчетов цифрового фильтра $\{h_n\}$ с частотной характеристикой аппроксимируемого им непрерывного фильтра $H(f)$. Реальный фильтр всегда будет иметь частотную характеристику, ограниченную в интервале частот от 0 до F :

$$H_y(f) = H(f) \operatorname{rect} \frac{f + F}{2F},$$

где функция rect равна единице вплоть до значения аргумента, равного единице, когда $f = F$.

Отсчеты импульсной реакции фильтра можно рассчитать через обратное преобразование Фурье

$$h_n = \int_{-\infty}^{\infty} H_y(f) \exp(j2\pi f \frac{n}{2F}) df = \int_{-F}^F H(f) \exp(j2\pi f \frac{n}{2F}) df.$$

Число отсчетов цифрового фильтра, которые закладывают в память машины для построения цифрового фильтра, всегда ограничено. Это вносит дополнительные искажения в частотную характеристику, получающегося цифрового фильтра.

Рассмотрим пример синтеза цифрового фильтра по заданной частотной характеристике непрерывного фильтра. Пусть требуется синтезировать фильтр с амплитудно-частотной характеристикой $H(f) = \exp\left(-\left(\frac{f}{F_0}\right)^2\right)$, где $F_0 = 200$ Гц (рис. 1.7). На первом этапе выбираем число отсчетов импульсной реакции, которое мы можем себе позволить. Например, примем число отсчетов равное $(2N + 1) = 5$. Далее, организуем симметричную частотную характеристику, дополнив ее зеркальным отражением в области отрицательных частот. Отсчеты частотной характеристики

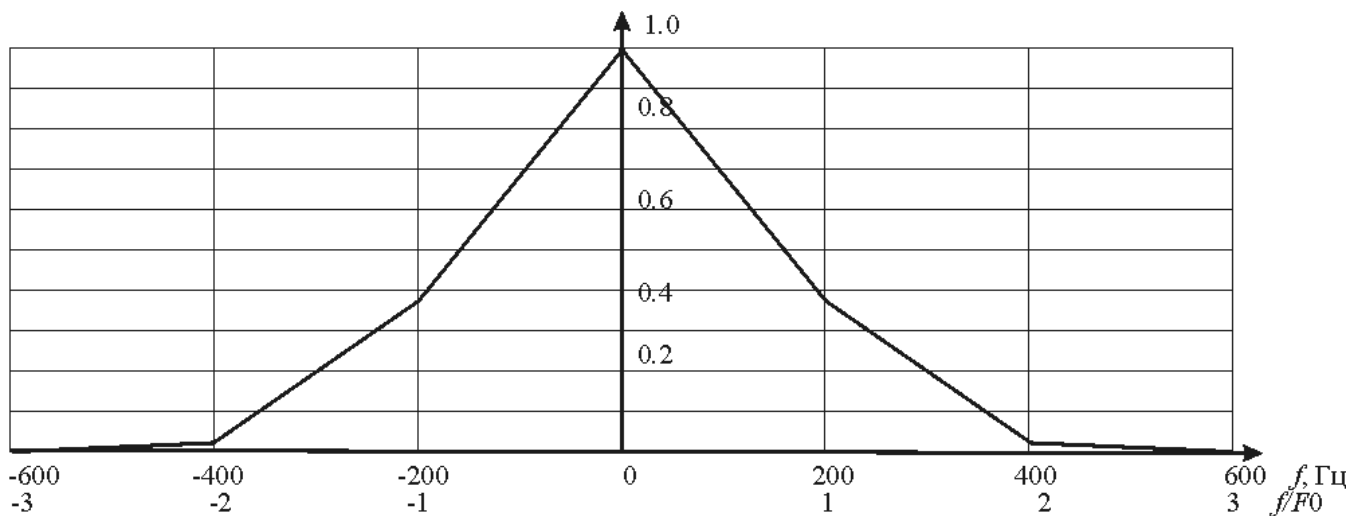


Рис. 1. 7. Симметричная частотная характеристика

$$H(0) = \exp(0) = 1; \quad H(1) = H(-1) = \exp(-1) = 0.368; \quad H(2) = H(-2) = \exp(-4) = 0.0183.$$

Затем выполняем обратное ДПФ:

$$h(kT) = \frac{1}{2N+1} \sum_{n=-N}^N K(nF_0) \exp\left(jkn \frac{2\pi}{2N+1}\right) = \frac{1}{2N+1} \sum_{n=-N}^N K(nF_0) \cos\left(kn \frac{2\pi}{2N+1}\right),$$

где T - интервал дискретизации, равный $1/F_0$; косинус появился потому, что мы организовали четную функцию и коэффициенты при синусах равны нулю.

В результате расчета: $h(0) = \frac{1 + 0.368 + 0.0183}{5} = 0.355$, так как $\cos(0) = 1$;
 $h(T) = 0.239$; $h(2T) = 0.083$.

После нормировки маска фильтра принимает вид:

0,234	0,673	1	0,673	0,234
-------	-------	---	-------	-------

Приведем формулу, позволяющую рассчитать двумерную пространственно-частотную характеристику (двумерный аналог АЧХ) в случае, когда матрица фильтра симметрична, как относительно среднего столбца и средней строки, так и относительно обеих главных диагоналей

$$H(2\pi f_x, 2\pi f_y) = a_0 + 2a_1(\cos 2\pi f_x + \cos 2\pi f_y) + 2a_2[\cos 2\pi(f_x + f_y) + \cos 2\pi(f_x - f_y)]$$

где маска фильтра

$$\begin{bmatrix} a_2 & a_1 & a_2 \\ a_1 & a_0 & a_1 \\ a_2 & a_1 & a_2 \end{bmatrix},$$

причем f_x, f_y – пространственные частоты по горизонтальному и вертикальному направлениям, которые измеряют в долях пространственной частоты дискретизации (т. е. величины, обратной расстоянию между соседними отсчетами) по соответствующей координате, и каждая из них изменяется в диапазоне частот от 0 до 0,5.

1.3. Препарирование изображений

Цифровые фильтры используют не только для улучшения изображения, а также для его *препарирования* с целью извлечения видеоданных. Типичным примером является *выделение контуров* изображения, полезных для распознавания образов. Различают три вида контуров: вертикальные, горизонтальные и наклонные. Для каждого из этих видов существуют фильтры, которые особенно хорошо выделяют соответствующие контуры. Фильтры, используемые для этих целей, могут быть *линейными*, *нелинейными* и *морфологическими*. Вертикальные перепады находят дифференцированием по строкам, горизонтальные – по столбцам. Дифференцирование выполняют цифровым способом, заменяя производные конечными приращениями:

$dL/dx \approx (L(k,l) - L(k-1,l))/\Delta x$; $dL/dy \approx (L(k,l) - L(k,l-1))/\Delta y$, где обычно принимают $\Delta x = \Delta y = 1$. Для выделения перепадов используют следующие наборы масок весовых множителей, реализующих двумерное дифференцирование:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}; \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}; \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

где направления склонов перепада условно обозначают «север», «восток», «юго-восток» и т. д.

Сумма весовых множителей масок равна нулю, что обеспечивает нулевой отклик на ровном участке изображения. Оператор Лапласа позволяет вычислить вторые производные от яркости изображения в двух направлениях. Маска для вычисления оператора Лапласа имеет вид:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

В этом случае по-прежнему сумма весовых коэффициентов равна нулю.

Увеличение центрального элемента маски на единицу означает добавление исходного изображения к контуру, выделенному оператором Лапласа. В результате будет повышена четкость исходного изображения.

Вектор, составленный из частных производных по вертикали и горизонтали, называют вектором-градиентом $grad(L(x, y))$. Интересует норма вектора градиента, объединяющая значения обеих производных. Тогда контур будет белым на черном фоне. Для облегчения вычислений используют норму l_1 , которая представляет собой сумму абсолютных значений производных. Такой оператор теряет свойство линейности: это означает, что даже теоретически вернуться к исходному изображению будет невозможно!

Для выделения, соответственно, вертикальных, горизонтальных и наклонных контуров (рис. 1.8) применяют маски, известные под названием «операторов Собела»:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}.$$

Оператор Собела, извлекающий одновременно все контура, имеет вид:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & j \end{bmatrix},$$

где $Sob = |(a + 2b + c) - (g + 2h + e)| + |(a + 2d + g) - (c + 2f + j)|$, причем буквами обозначены значения яркости в элементах битовой матрицы, охваченных маской фильтра.

Другим методом выделения контуров изображения является *нерезкое маскирование*: из исходного изображения вычитают его сглаженную копию (рис. 1.9). Полученный контурный препарат представляет самостоятельную ценность, а также может быть добавлен с определенным весом к исходному изображению. В результате будет повышена его резкость.



Рис. 1.8. Результат выделения наклонных контуров

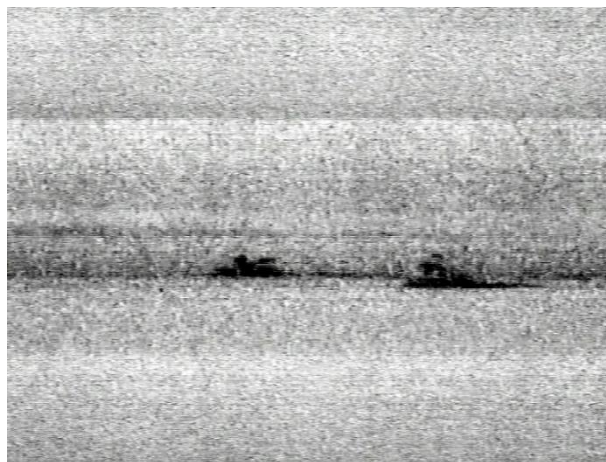


Рис. 1.9. Препарат нерезкого маскирования

1.4. Медианная фильтрация

Для обработки изображений широко применяют нелинейную медианную фильтрацию. Рангово-порядковые (медианные) фильтры относят к отдельному классу сглаживающих фильтров, которые позволяют эффективно подавлять точечные помехи и шумы, сохраняя при этом резкие перепады яркости на границах объектов. Однако такой фильтр может подавить в изображении мелкие детали, размеры которых меньше маски фильтра. Предположим, что строка битовой матрицы выглядит следующим образом: 17 200 55 64 100 99 По-видимому, здесь имеется грубая ошибка во втором элементе, который резко отличается от соседних элементов и выглядит совершенно изолированным. Рангово-порядковые фильтры не изменяют отдельные значения яркости, а производят их сортировку в пределах маски. Сильно отличающийся от соседей элемент заменяют средним значением из числа сортируемых элементов. Проведем эту сортировку: 17 55 200; 55 64 200; 55 64 100; 64 99 100 ... , в результате получим новую фильтрованную последовательность: 55 64 64 99, где аномальный элемент безвозвратно устранен.

Аналогичным образом выполняют медианную фильтрацию двумерных изображений: в фокус маски (например, 3×3) ставят, после сортировки элементов в пределах маски, средний по рангу элемент. В результате медианной фильтрации с маской 3×3 на нижнем изображении (рис. 1.10) среднеекватическое



Исходное изображение



Рис. 1.10. Результат медианной фильтрации

значение шума заметно снизилось по сравнению с исходным верхним изображением.

Медианный фильтр сохраняет перепад: если по одну стороны границы все точки сохраняют свои значения: 5;5;5;5;5;1;1;1;1;. После прохождения медианного фильтра с апертурой в три точки мы получим следующую последовательность: 5;5;5;5;1;1;1.

Формально медианный фильтр описывают так:
 $y_i = \text{mediana}(x_i) = \text{mediana}(x_{i-v}, \dots, x_i, \dots, x_{i+v}),$
где $v = (n-1)/2$, а $n = (2j+1)$ - нечетное число, чтобы у маски был центральный элемент (фокус).

Шум с нормальным законом распределением медианный фильтр подавляет примерно на 50% хуже, чем усредняющий фильтр. Зато не искажает перепады яркости в исходном изображении. Медианный фильтр сохранит верное значение сигнала в том и только в том случае, если число выбросов m в пределах аперттуры фильтра длиной N меньше половины ее ширины, т. е. $m < (n-1)/2$. Это свойство сохраняется и для двумерной аперттуры.

Имеются модификации медианного фильтра: взвешенный, адаптивный, с детектором импульсов [6].

1.5. Цели и задачи лабораторной работы

Целью лабораторной работы является практическая реализация и анализ методов фильтрации цифровых изображений. В соответствии с заданием на моделирование необходимо разработать программу в виде консольного приложения, реализующую следующие функции:

1. Загрузку растрового изображения и вывод его на экран.
2. Фильтрацию изображений различными фильтрами, указанными в таблице 1.1.
3. Получение контурных препаратов изображения с помощью алгоритма DoG, детектора границ Кэнни, а также фильтров выделения контуров в соответствии с индивидуальным заданием, варианты которого приведены в таблице 1.2.

1.6. Фильтрация изображений

В лабораторной работе предлагается реализовать набор фильтров, позволяющих добиться на изображении определенного эффекта, например, сглаживания, повышения резкости, выделения контуров. Для того, чтобы добиться сглаживания, можно применить операцию свертки изображения с маской, где коэффициенты распределены по двумерной гауссиане. На каждой итерации яркость пикселя, соответствующего фокусу (центральному элементу) маски, заменяется рассчитанным значением свертки, поделенным на нормировочный коэффициент, величина которого равна сумме элементов маски (см. рис. 1.11). При этом значения граничных пикселей как правило не изменяются.

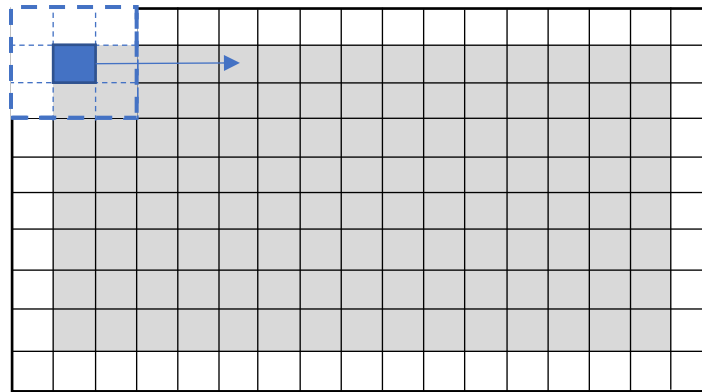


Рис. 1.11. Двумерная фильтрация.

Рассмотрим программную реализацию сглаживающего фильтра:

```
void gauss3(const Mat &input_img, Mat &output_img)
{
    output_img = Mat::zeros(input_img.size(), CV_8U);
    float k = 36; // коэффициент нормировки
    float Fk[3][3] = { {1,4,1}, {4,16,4}, {1,4,1} }; // маска фильтра
    for (int i = 1; i < input_img.cols - 1; i++)
        for (int j = 1; j < input_img.rows - 1; j++) {
            uchar pix_value = input_img.at<uchar>(j, i);
            // далее производим свертку
            float Rez = 0;
            for (int ii = -1; ii <= 1; ii++)
                for (int jj = -1; jj <= 1; jj++) {
                    uchar blurred = input_img.at<uchar>(j + jj,
i + ii);
                    Rez += Fk[ii + 1][jj + 1] * blurred;
                }
            output_img.at<uchar>(j, i) = Rez / k;
        }
```

```

    }
    uchar blurred = Rez / k; // осуществляем нормировку
    output_img.at<uchar>(j, i) = blurred;
}
}

```

Функция принимает на вход полутоновое изображение `input_img` и возвращает размытое `output_img`. Обратите внимание на границы итерирования в циклах `for()`. Результат работы данной функции представлен на рис. 1.12. Эффект размытия выражен достаточно слабо – если требуется более сильное размытие, можно увеличить апертуру фильтра.

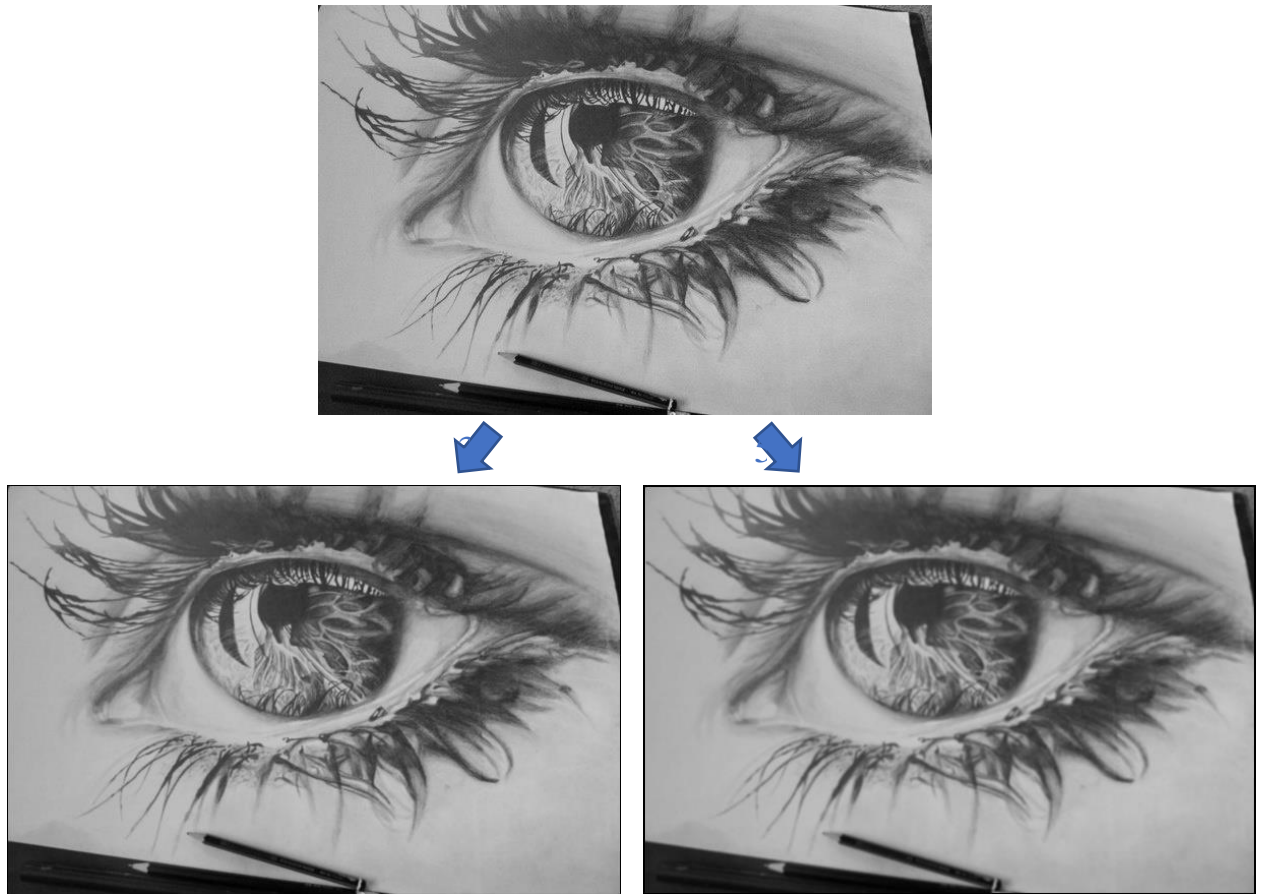


Рис. 1.12. Сглаживающий фильтр.

Оставшиеся фильтры из таблицы 1.1 предлагается реализовать самостоятельно с использованием описания, приведенного в теоретической части лабораторной работы.

Таблица 1.1. Задание на фильтрацию изображений

	Вид фильтра	Размер апертуры
1	Фильтр Гаусса	3
2	Фильтр Гаусса	5
3	Фильтр «Мозаика»	3
4	Фильтр апертурный коррекции	3
5	Медианный фильтр	3

1.7. Получение контура изображений

Результат фильтрации, полученный на предыдущем этапе работы, можно использовать для получения контурного препарата. Для этого достаточно попиксельно взять модуль разности между откликами фильтров Гаусса с апертурой 3 и 5. Такой вид фильтрации называют «разность гауссианов» (*Difference of gaussian, DoG*):

$$DoG(x, y) = G_{\sigma_1}(x, y) - G_{\sigma_2}(x, y) = \frac{1}{\sqrt{2\pi}} \left(\frac{1}{\sigma_1} e^{-(x^2+y^2)/2\sigma_1^2} - \frac{1}{\sigma_2} e^{-(x^2+y^2)/2\sigma_2^2} \right).$$

Модуль разности между двумя изображениями можно найти с помощью функции `cv::absdiff(img1, img2, diff)`. На рисунке 1.3 (а) приведен результат выделения контуров с помощью алгоритма *DoG*. Для большей наглядности все значения контурного препарата были увеличены в 7 раз. В OpenCV это можно сделать одним оператором умножения: `diff * 7`.

На практике для выделения контуров очень часто используют детектор границ Кэнни. Реализация алгоритма достаточно сложна и включает в себя следующие этапы:

- сглаживание для подавления шумов;
- поиск градиентов, которые и будут определять найденные границы;
- подавление немаксимумов для того, чтобы сузить границы до 1 пикселя;
- двойная пороговая фильтрация, в результате которой пиксели, значение которых на карте градиентов меньше заданного нижнего порога относятся к фону, а выше верхнего порога – к границам;

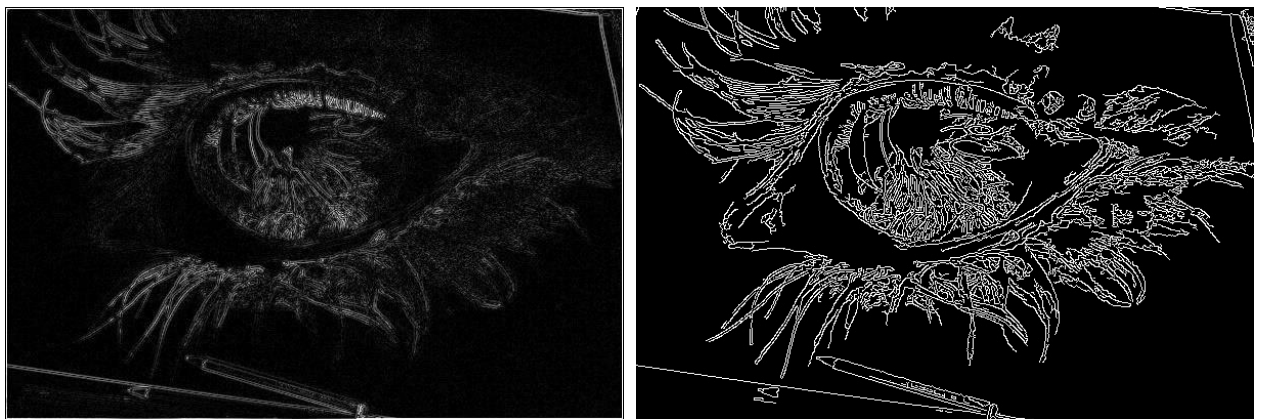
- трассировка областей неоднозначности, заключающая в уточнении границ в областях, где значение градиентов находится в интервале между заданными порогами.

Библиотека OpenCV позволяет применять фильтр Кэнни с помощью функции `cv::Canny(img, contour, thr1, thr2)`, где *thr1* и *thr2* – значения нижнего и верхнего порога соответственно. На рис. 1.13 (б) представлен результат применения детектора границ Кэнни с параметрами *thr1* = 50, *thr2* = 200.

Для выделения границ на изображении может использован расчет приближенного значения первой или второй производной. В таблице 1.2 приведены виды фильтров, один из которых требуется реализовать самостоятельно в соответствии с выбранным вариантом.

Таким образом, по результатам выполнения данного пункта должны быть реализованы следующие виды фильтров:

- разность гауссианов;
- фильтр Кэнни (используется готовая реализация OpenCV с эмпирически подобранными значениями порогов);
- индивидуальный вариант в соответствии с таблицей 1.2.



а

Рис. 1.13. Выделение границ на изображении (а – разность гауссианов, б – детектор границ Кэнни).

Таблица 1.2. Индивидуальное задание на получение контура изображения

Номер варианта	Вид фильтра
1	Оператор Собела для горизонтальных контуров
2	Оператор Собела для вертикальных контуров

3	Оператор Собела для наклонных контуров
4	Обобщенный оператор Собела
5	Оператор Превитта для горизонтальных контуров
6	Оператор Превитта для вертикальных контуров
7	Оператор Робертса для горизонтальных контуров
8	Оператор Робертса для вертикальных контуров
9	Оператор Шарра
10	Оператор Лапласа

1.8. Содержание отчета

В отчете должны быть приведены:

1. Цель работы.
2. Краткие теоретические сведения по теме работы.
3. Исходное полутоновое изображение и результат его обработки с помощью фильтров, указанных в таблице 1.1.
4. Контурные изображения исходного полутонового изображения, найденные с помощью алгоритма «разность гауссианов», детектора границ Кэнни, а также выбранного фильтра из таблицы 1.2.
5. Выводы об особенностях и результатах применения фильтров, реализованных в лабораторной работе.

1.9. Контрольные вопросы

1. Поясните разницу между линейными, нелинейными и логическими цифровыми фильтрами. Приведите примеры фильтров каждого класса.
2. В чем отличие БИХ и КИХ фильтров?
3. Что такое импульсная реакция фильтра?
4. Сравните операцию свертки в сигнальной и спектральной области.
5. Перечислите и поясните основные шаги синтеза цифрового фильтра.