

МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное учреждение высшего
профессионального образования
«Национальный исследовательский университет «МИЭТ»

Кафедра высшей математики № 1

Алимагадов Курбан Алимагадович

Бакалаврская работа
по направлению 01.03.04 «Прикладная математика»

Исследование эффективности пространственных и частотных методов
фильтрации шумов на изображениях

Студент _____ (Алимагадов К. А.)

Научный руководитель

д.ф.-м.н., профессор _____ (Умняшкин С. В.)

Аннотация

Работа посвящена исследованию эффективности применения пространственных и частотных методов фильтрации цифровых изображений, чтобы определить, какие из них лучше решают задачу подавления шумов.

Вследствие несовершенства электронных устройств, с помощью которых человек получает и передаёт цифровые изображения на них образуются помехи и шумы. Причинами возникновения шума могут быть воздействия друг на друга элементов светочувствительной матрицы снимающего устройства, перепады токов в нём и т. д. Большое количество современных программных и технических средств используют информацию, получаемую с изображений, для принятия различных решений на её основе. Наличие шума на изображениях влияет на качество работы этих технологий.

Для подавления шумов применяются методы фильтрации, использующие различные подходы к обработке изображений. При выборе конкретного фильтра, который планируется использовать для устранения шума, важно знать, как эффективно он будет справляться с этой задачей, и какое количество вычислительных операций он будет при этом выполнять.

Поэтому на сегодняшний день определение эффективности методов фильтрации является очень актуальным, так как позволит дать рекомендации по их применению в различных областях науки и техники.

В работе были рассмотрены 7 методов фильтрации нормального белого шума: среднеарифметический фильтр, фильтр срединной точки, фильтр Гаусса, билатеральный фильтр, “жесткий” и “мягкий” пороговые фильтры, и фильтр Винера. А также 2 метода фильтрации периодического шума: фильтр Винера и метод оптимальной узкополосной фильтрации, причём для второго было рассмотрено две его модификации, использующие разные функции для выделения шумовой составляющей. Представлены результаты применения пороговых фильтров для обработки ДПФ зашумлённых изображений, приведены значения метрик, позволяющие оценить уровень зашумлённости изображения, а также качество визуального восприятия до и после фильтрации.

Оглавление

Введение	4
Глава 1. Теоретические основы	6
1.1. Изображение и модели шума	6
1.2. Методы фильтрации	7
1.2.1. Пространственные фильтры	7
1.2.1.1. Среднеарифметический фильтр	9
1.2.1.2. Фильтр срединной точки	10
1.2.1.3. Фильтр Гаусса	10
1.2.1.4. Билатеральный фильтр	11
1.2.2. Частотные фильтры	12
1.2.2.1. “Жёсткий” и “мягкий” пороговые фильтры	12
1.2.2.2. Фильтр Винера	13
1.2.2.3. Метод оптимальной узкополосной фильтрации	14
1.3. Способы оценки эффективности фильтрации	15
1.4. Выводы	17
Глава 2. Особенности и детали реализации исследуемых методов	18
2.1. Оценка параметров шума на изображении	18
2.2. Реализация методов “жёсткой” и “мягкой” пороговой фильтрации применимых к обработке ДПФ зашумлённого изображения	20
2.3. Реализация фильтра Винера для фильтрации нормального белого шума	22
2.4. Реализация фильтра Винера для фильтрации периодического шума	23
2.5. Скользящее окно и весовая маска	24
2.6. Выводы	27
Глава 3. Тестирование фильтров, анализ результатов и выводы	28
3.1. Результаты фильтрации тестовых изображений	28
3.2. Вычислительная сложность исследуемых методов фильтрации	34
3.3. Выводы	36
Заключение	37
Список источников	38
Приложения	39

Введение

Процессы получения и передачи цифровых изображений сопровождаются образованием на них искажений и шумов. Шум представляет собой дефекты на изображении, возникновение которых зависит от множества факторов: качество сенсоров фото- или видеокамеры, освещённость сцены, температура сенсоров, изменения токов, протекающих в устройстве, с помощью которого человек получает изображение [1, стр. 368].

Фильтрация шумов на изображениях – одна из важных задач цифровой обработки сигналов, часто возникающая во многих инженерных областях человеческой деятельности.

Среди фильтров различают пространственные, работающие непосредственно с пикселями изображения, и частотные, которые используют дискретное преобразование Фурье (ДПФ) изображения и основываются на изменении Фурье-образа с последующим применением обратного ДПФ [2, стр. 44, 78]. На сегодняшний день существует большое количество методов, применяющих эти два подхода.

В связи с этим возникает необходимость узнать, какие фильтры лучше справляются с задачей подавления шумов на изображениях. Это позволило бы дать инженерам и программистам рекомендации, помогающие при выборе того или иного метода фильтрации в процессе создания программ и электронных устройств, решающих эту задачу.

Для оценки качества работы фильтров можно использовать следующие метрики: отношение сигнал/шум “Signal-to-Noise Ratio”, пиковое отношение сигнал/шум “Peak Signal-to-Noise Ratio” и индекс структурного сходства “Structural Similarity Index” [1, стр. 416], [3].

Цель работы: исследовать эффективность применения рассматриваемых методов фильтрации и определить, какие из них лучше решают задачу подавления шумов на изображениях.

Задачи, решаемые в рамках работы:

- 1) Изучить существующие пространственные и частотные методы фильтрации нормального и периодического шумов на изображениях;

- 2) Рассмотреть применение методов “мягкой” и “жёсткой” фильтрации, используемых обычно в базисе вейвлетов, к ДПФ изображения, зашумлённого белым гауссовым шумом;
- 3) Сравнить эффективность работы рассматриваемых методов, анализируя результаты фильтрации зашумлённых тестовых изображений;
- 4) Сделать выводы и дать рекомендации на основе полученных данных.

Актуальность работы заключается в том, что на сегодняшний день эффективное решение задачи устранения шумов является одним из необходимых условий надёжной и качественной работы систем обработки и передачи изображений во многих сферах деятельности человека (фото- и видеосъёмка, телевидение, компьютерное зрение, автоматизированные системы управления и т. д.).

В первой главе проводится обзор теории, рассматриваются модели нормального белого и периодического шумов, существующие алгоритмы и методы фильтрации, а также метрики, используемые для оценки уровня шума на изображении и улучшения его визуального качества после применения фильтров.

Во второй главе описываются особенности реализации фильтров, представленных в работе, приводятся алгоритмы “мягкой” и “жёсткой” пороговой фильтрации применимые к обработке ДПФ зашумлённого изображения.

Третья глава содержит результаты работы фильтров, полученные на компьютере, их анализ, выводы и рекомендации.

Глава 1. Теоретические основы

1.1. Изображение и модели шума

Под монохроматическим (или полутонным) изображением будем понимать функцию $f(x, y)$ координат (x, y) точек плоскости, значение которой интерпретируется как яркость точки изображения с координатами (x, y) .

Для цифровой обработки, передачи и хранения изображения $f(x, y)$ его преобразуют в цифровую форму, т.е. выполняют дискретизацию и квантование этого изображения. Дискретизация – это замена изображения $f(x, y)$, заданного в прямоугольнике $0 \leq x \leq A, 0 \leq y \leq B$, матрицей его значений в узлах прямоугольной сетки:

$$\{f_{m,n}\} = \{f(m\Delta x, n\Delta y)\}; \quad m = 0, 1, \dots, M-1, \quad n = 0, 1, \dots, N-1, \quad (1)$$

где $\Delta x = A/M$ и $\Delta y = B/N$ – шаги дискретизации. Квантование – замена значений яркости дискретного изображения $\{f_{m,n}\} = \{f(m\Delta x, n\Delta y)\}$ числами из конечного множества $\Omega = \{d_j\}_{j=0}^{K-1}$ возможных уровней яркости. При этом число L уровней (градаций) яркости обычно выбирают равным натуральной степени двойки, т. е. $L = 2^k$, а дискретные уровни яркости полагают равными целым числам из промежутка $[0; L-1]$ [2, стр. 7, 16-17].

В работе рассмотрены две аддитивные модели шумов, часто появляющихся на изображениях: нормальный (гауссов) шум и периодический шум. При возникновении аддитивных шумов шумовая компонента $n(x, y)$ прибавляется к изображению $f(x, y)$:

$$g(x, y) = f(x, y) + n(x, y), \quad (2)$$

где $g(x, y)$, $n(x, y)$ и $f(x, y)$ – соответственно зашумлённое изображение, шум и исходное изображение [1, стр. 367-368].

Нормальный шум появляется на изображениях в результате воздействия множества факторов (перепады напряжения, паразитные токи, взаимное влияние элементов светочувствительной матрицы снимающего устройства). Для модели такого шума $n(x, y) = z$ – случайная величина, плотность распределения которой задаётся законом Гаусса:

$$\rho(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, \quad (3)$$

где μ – математическое ожидание, σ – среднеквадратическое отклонение [2, стр. 92].

В рамках работы будем рассматривать белый гауссов шум, то есть такой, у которого спектральная плотность является константой и равномерна распределена по всем частотам [1, стр. 369].

Причиной появления периодического шума обычно являются электрические или электромеханические помехи при получении изображения. Моделью периодического шума является двумерное пространственное гармоническое колебание, которое задаётся формулой:

$$r(x, y) = A * \cos \left[2\pi \left(\frac{u_0}{M} x + \frac{v_0}{N} y \right) + \gamma \right], \quad (4)$$

где $A \geq 0$ – амплитуда, числа u_0 и v_0 определяют частоты по осям x и y соответственно, а γ – начальную фазу колебания [2, стр. 94].

1.2. Методы фильтрации

Для решения задачи фильтрации изображений применяются два вида фильтров: пространственные и частотные.

В данной работе для фильтрации белого нормального шума применяются 4 пространственных фильтра: среднеарифметический фильтр, фильтр срединной точки, фильтр Гаусса и билатеральный фильтр, – и 3 частотных фильтра: “жесткий” пороговый фильтр, “мягкий” пороговый фильтр и фильтр Винера. Для фильтрации периодического шума используются 2 частотных метода: оптимальной узкополосной фильтрации и фильтр Винера.

1.2.1. Пространственные фильтры

Процессы пространственной обработки описываются уравнением

$$g(x, y) = T[f(x, y)], \quad (5)$$

где $f(x, y)$ – исходное изображение, $g(x, y)$ – обработанное изображение, а T – оператор, действующий на $f(x, y)$.

Если оператор, применяемый к пикселям исходного изображения, является линейным, то фильтр называют линейным пространственным фильтром. В этом случае его можно задать матрицей, которую также называют маской, ядром или окном. Фильтрация создаёт новые пиксели, значения яркости которых зависят от значений весов маски, а координаты совпадают с координатами центра окрестности.

Обработанное (фильтрованное) изображение возникает в процессе прохода фильтра по изображению окном и вычисления свёртки значений пикселей, попавших в окрестность, с весовой маской – ядром свёртки [1, стр. 186-188], [2, стр. 44]. На рисунках 1 и 2 изображены соответственно схемы обрабатываемой окрестности и маски размера 3×3 .

$g(x - 1, y - 1)$	$g(x - 1, y)$	$g(x - 1, y + 1)$
$g(x, y - 1)$	$g(x, y)$	$g(x, y + 1)$
$g(x + 1, y - 1)$	$g(x + 1, y)$	$g(x + 1, y + 1)$

Рис. 1. Схема обрабатываемой окрестности

$w(-1, -1)$	$w(-1, 0)$	$w(-1, 1)$
$w(0, -1)$	$w(0, 0)$	$w(0, 1)$
$w(1, -1)$	$w(1, 0)$	$w(1, 1)$

Рис. 2. Схема весовой маски

В общем виде фильтрация изображения размерами $M \times N$ линейным пространственным масочным фильтром размерами $m \times n$, $m = 2a + 1$, $n = 2b + 1$ (a , b – целые неотрицательные числа), задаётся формулой:

$$g(x, y) = \sum_{s=-(m-1)/2}^{(m-1)/2} \sum_{t=-(n-1)/2}^{(n-1)/2} w(s, t) f(x + s, y + t), \quad (6)$$

где $w(s, t)$ – весовая маска, $g(x, y)$ – преобразованное изображение, $\hat{f}(x, y)$ – фильтруемое изображение [2, стр. 59-60].

Для нелинейных фильтров требуется, чтобы был задан размер окрестности и операция (операции), выполняющиеся над значениями пикселей изображения, попадающими в окрестность [1, стр. 194].

При применении методов фильтрации особое внимание уделяется обработке пикселей вблизи границ изображения, так как окрестность обработки здесь может выходить за пределы изображения (см. рис. 3). В этом случае можно полагать, что отсчёты двумерного сигнала $f(x, y)$ при $x > M$ и $y > N$ равны либо нулю, либо значениям, которые $f(x, y)$ принимает на границе [4, стр. 112].

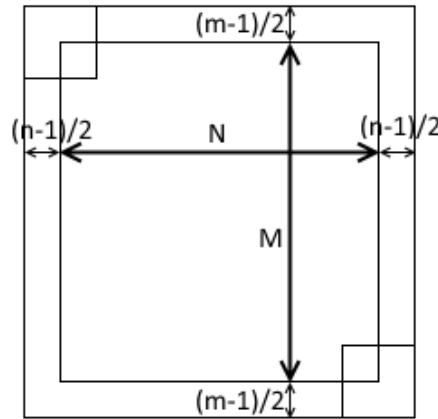


Рис. 3. Схема фильтрации изображения масочным фильтром

1.2.1.1. Среднеарифметический фильтр

Среднеарифметический фильтр является линейным. Пусть S_{xy} обозначает прямоугольную окрестность (множество координат точек изображения) размерами $m \times n$ с центром в точке (x, y) . Значение восстановленного изображения \hat{f} в произвольной точке (x, y) представляет собой среднее арифметическое значений в точках, принадлежащих окрестности S_{xy} , зашумлённого изображения $g(x, y)$:

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t). \quad (7)$$

Эта операция может быть реализована в виде пространственного фильтра размерами $m \times n$, в котором все коэффициенты равны $1/mn$. Усредняющий фильтр сглаживает вариации яркости на изображении, и в результате этого сглаживания происходит уменьшение шума [1, стр. 378-379].

Среднеарифметический фильтр решает задачу устранения белого нормального шума, но вместе с тем он сильно размывает изображение.

1.2.1.2. Фильтр срединной точки

Фильтр срединной точки является нелинейным. Его применение заключается в замене значения яркости очередного пикселя на среднее арифметическое между максимальным и минимальным значениями в соответствующей окрестности:

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right], \quad (8)$$

S_{xy} – окрестность, в пределах которой происходит поиск минимума и максимума, $g(x, y)$ – фильтруемое изображение, $\hat{f}(x, y)$ – отфильтрованное изображение [1, стр. 383].

Данный фильтр использует усреднение, а также учитывает порядковые статистики и хорошо решает задачу фильтрации белого нормального шума.

1.2.1.3. Фильтр Гаусса

Фильтр Гаусса является линейным. Весовая маска формируется на основе гауссовой функции двух переменных

$$h(x, y) = e^{-\frac{x^2+y^2}{2\sigma_D^2}}, \quad (9)$$

где σ_D – стандартное отклонение, координаты точек x, y на плоскости полагаются целыми числами. Чтобы сформировать из этой функции маску фильтра размерами $m \times n$, ($m = 2a + 1, n = 2b + 1$), берутся значения этой функции вокруг её центра

$$w(i, j) = \frac{1}{S} \begin{bmatrix} h(-\frac{m-1}{2}, -\frac{n-1}{2}) & \dots & h(-\frac{m-1}{2}, \frac{n-1}{2}) \\ \vdots & \ddots & \vdots \\ h(\frac{m-1}{2}, -\frac{n-1}{2}) & \dots & h(\frac{m-1}{2}, \frac{n-1}{2}) \end{bmatrix}, \quad (10)$$

$$S = \sum_{i=-\frac{m-1}{2}}^{\frac{m-1}{2}} \sum_{j=-\frac{n-1}{2}}^{\frac{n-1}{2}} h(i, j), \quad (11)$$

S – нормировочный коэффициент (сумма всех используемых для формирования фильтра значений функции $h(x, y)$) [1, стр. 194]. Этот фильтр, как и среднеарифметический, устраняет шум, заменяя каждый пиксель на взвешенную сумму соседних. Однако он меньше размывает изображение, так как пиксели,

находящиеся ближе к обрабатываемому, вносят больший вклад в сумму значений яркости.

1.2.1.4. Билатеральный фильтр

Идея билатерального фильтра заключается в том, что при формировании весовой маски учитывается расстояние между пикселями из обрабатываемой окрестности, а также близость значений их интенсивностей. Большая разница между яркостями соседних пикселей обычно обуславливается не наличием шума, а локальными особенностями изображения: контурами и мелкими деталями. Поэтому пиксели, находящиеся поблизости, но сильно отличающиеся по яркости, при сглаживании не должны сильно влиять друг на друга.

Данный фильтр похож на фильтр Гаусса, однако билатеральная фильтрация учитывает сразу и расстояние между рассматриваемыми пикселями, и разницу между значениями их яркостей.

Формула для расчёта яркости пикселей отфильтрованного изображения имеет вид:

$$\hat{f}(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b g(x+s, y+t) w(s, t) h[g(x+s, y+t) - g(x, y)]}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) h[g(x+s, y+t) - g(x, y)]} \quad (12)$$

$$w(s, t) = \exp\left(-\frac{1}{2} \frac{s^2 + t^2}{\sigma_d^2}\right) \quad (13)$$

$$h(u) = \exp\left(-\frac{1}{2} \frac{u^2}{\sigma_r^2}\right), \quad (14)$$

где σ_d и σ_r – константы [5, стр. 1142].

Если удаётся хорошо подобрать константы σ_d и σ_r , билатеральный фильтр показывает отличные результаты, сильно уменьшая уровень шума и сохраняя многие детали изображения.

Недостатками метода являются сложность подбора параметров, а также большой объём вычислений.

1.2.2. Частотные фильтры

Фильтрация в частотной области заключается в изменении Фурье-образа изображения и последующем обратном преобразовании Фурье для получения обработанного результата. Это утверждение можно сформулировать в виде следующего основного уравнения частотной фильтрации:

$$\hat{f}(x, y) = \mathfrak{F}^{-1}[H(u, v)F(u, v)], \quad (15)$$

где $F(u, v)$ – ДПФ фильтруемого изображения $f(x, y)$ размерами $M \times N$, $H(u, v)$ – частотная характеристика фильтра, \mathfrak{F}^{-1} – обратное ДПФ, а $\hat{f}(x, y)$ – результат фильтрации (выходное изображение).

Низкие частоты в преобразовании связаны с областями медленно меняющейся яркости, такими как стены комнаты или безоблачное небо на изображении пейзажа. Высокие частоты, наоборот, соответствуют резким переходам яркости, таким как контуры и шумы. Таким образом, самой низкой частоте Фурье-образа изображения соответствует постоянная составляющая – коэффициент ДПФ, отвечающий за общую яркость и тональность изображения. Применяя частотные фильтры, следует уделять особое внимание тому, чтобы данный коэффициент не фильтровался, для предотвращения потери яркости восстановленного изображения [1, стр. 306-307].

1.2.2.1. “Жёсткий” и “мягкий” пороговые фильтры

Пороговую фильтрацию обычно используют в базисе вейвлетов для подавления коэффициентов разложения, соответствующих шумовой составляющей.

Алгоритмы вейвлет-фильтрации похожи на алгоритмы частотной фильтрации и включают три следующих этапа:

1. взятие прямого вейвлет-преобразования от зашумлённых значений функции;
2. обработка коэффициентов разложения;
3. взятие обратного вейвлет-преобразования от обработанных коэффициентов.

Одним из простейших алгоритмов обработки зашумлённых коэффициентов вейвлет-разложения является использование на втором этапе порогового алгоритма с “жёсткой” пороговой функцией [6, стр. 143, 157]:

$$hard(\tilde{d}, \lambda) = \begin{cases} \tilde{d}, & \text{если } |\tilde{d}| \geq \lambda; \\ 0, & \text{если } |\tilde{d}| < \lambda, \end{cases} \quad (16)$$

где \tilde{d} – значение коэффициента дискретного вейвлет-преобразования, λ – заданный порог.

“Мягкая” пороговая фильтрация задаётся функцией [6, стр. 158]:

$$soft(\tilde{d}, \lambda) = \begin{cases} sign(\tilde{d}) \cdot [|\tilde{d}| - \lambda], & \text{если } |\tilde{d}| \geq \lambda; \\ 0, & \text{если } |\tilde{d}| < \lambda. \end{cases} \quad (17)$$

Аналогичные подходы в рамках исследования были применены к фильтрации ДПФ зашумлённых изображений и подробно описаны в главе 2.

1.2.2.2. Фильтр Винера

Фильтр Винера должен обеспечивать формирование наилучшей оценки полезного сигнала $\hat{f}(x, y)$ в смысле минимума мощности сигнала-ошибки $E(x, y) = f(x, y) - \hat{f}(x, y)$ [7, стр. 307]:

$$\sigma_E^2 = M \left\{ \left(f(x, y) - \hat{f}(x, y) \right)^2 \right\} \rightarrow \min. \quad (18)$$

Частотная характеристика фильтра задаётся выражением [1, стр. 414]:

$$W(u, v) = \frac{1}{H(u, v)} \cdot \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)}, \quad (19)$$

где $H(u, v)$ – искажающий оператор, $S_\eta(u, v)$ – спектр мощности шума, $S_f(u, v)$ – спектр мощности исходного изображения.

В случае фильтрации только шума искажение отсутствует ($H(u, v) = 1$) и выражение принимает вид

$$W(u, v) = \frac{S_f(u, v)}{S_f(u, v) + S_\eta(u, v)}. \quad (20)$$

Так как спектры мощности шума и исходного изображения, как правило, нам изначально неизвестны, вместо них используются их приближённые оценки $\hat{S}_\eta(u, v)$ и $\hat{S}_f(u, v)$ [7, стр. 309] [1, стр. 414-415]. Способы определения этих оценок, используемые в рамках данной работы, будут подробно разобраны в следующей главе.

1.2.2.3. Метод оптимальной узкополосной фильтрации

Данный метод является оптимальным в том смысле, что он минимизирует значения локальной дисперсии восстановленного изображения $\hat{f}(x, y)$. Метод состоит в том, чтобы сначала получить в виде отдельного изображения основной вклад, приносимый помехой, а затем вычесть из исходного искажённого изображения некоторую непостоянную весовую долю, полученного изображения помехи [1, стр. 399].

Первый шаг состоит в выделении основных частотных составляющих помехи. Далее мы исходим из предположения, что искажённое изображение $g(x, y)$ получается из неискажённого изображения $f(x, y)$ прибавлением помехи $\eta(x, y)$. Фильтрация позволяет получить лишь некоторое приближение к функции, определяющей шумовую составляющую. Эффект, связанный с отличием построенного приближения $\eta(x, y)$ от реально существующей помехи, может быть уменьшен, если при построении приближения для неискажённого изображения $f(x, y)$, мы вычтем из искажённого изображения $g(x, y)$ некоторую взвешенную долю функции $\eta(x, y)$:

$$\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y), \quad (21)$$

где $w(x, y)$ – подлежащая определению весовая функция.

Поиск функции $w(x, y)$, обеспечивающей минимум локальной дисперсии восстановленного изображения $\hat{f}(x, y)$, представляет собой задачу нахождения экстремума (минимума), решение которой [1, стр. 401]:

$$w(x, y) = \frac{\overline{g(x, y)w(x, y)} - \bar{g}(x, y)\bar{w}(x, y)}{\overline{\eta^2(x, y)} - \bar{\eta}^2(x, y)}, \quad (22)$$

где верхнее подчеркивание означает операцию вычисления среднего арифметического по некоторой заданной области: $(2a + 1) \times (2b + 1)$. Фильтр проходит по изображению скользящим окном, размер которого равен размеру области, в пределах которой мы находим минимум локальной дисперсии.

Рассмотрим фильтры, использующиеся в работе для выделения частотных составляющих помехи. Для решения этой задачи применялись фильтр Гаусса и фильтр Баттерворта.

Фильтр Гаусса определяется формулой [1, стр. 327]:

$$G(u, v) = e^{-\frac{D^2(u, v)}{2\sigma_G^2}}, \quad (23)$$

где σ_G задаёт размах вокруг центра (экстремума функции $G(u, v)$), а $D(u, v)$ равно:

$$D(u, v) = \sqrt{(u - u_0)^2 + (v - v_0)^2}, \quad (24)$$

u_0 и v_0 – координаты экстремума гауссова импульса.

Фильтр Баттерворта определяется формулой:

$$B(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{\sigma_B}\right)^{2n}}, \quad (25)$$

σ_B задаёт размах вокруг центра, n – порядок фильтра [1, стр. 328].

1.3. Способы оценки эффективности фильтрации

Описанные в этом разделе величины, используемые для оценки эффективности фильтрации, в инженерной литературе называются метриками, хотя в строго математическом смысле этого слова они таковыми не являются.

Для оценки уровня зашумлённости сигнала в научных исследованиях обычно используется величина, называемая отношением сигнал-шум (Signal-to-Noise Ratio или SNR), которая равна отношению мощности полезного сигнала к мощности шума или квадрату отношений амплитуд сигнала и шума. SNR, как правило, охватывает большой диапазон значений, и часто интерес представляет не точное значение отношения, а его порядок. Поэтому принято рассматривать 10 десятичных логарифмов этого отношения и выражать SNR в децибелах:

$$SNR_{дБ} = 10 \lg \left(\frac{A_s}{A_n} \right)^2 = 20 \lg \frac{A_s}{A_n}, \quad (26)$$

где A_s и A_n соответственно среднеквадратические значения амплитуд сигнала и шума, рассчитанные по всем отсчётам (пикселям) изображения [1, стр. 416].

При сравнении методов подавления шума на изображениях, а также методов сжатия изображений, традиционно используется величина PSNR (Peak Signal-to-Noise Ratio) – пиковое отношение сигнал-шум. В отличие от SNR, PSNR является отношением максимума возможного значения сигнала к значению шума, а не фактического значения полезного сигнала к значению шума. По определению, PSNR между изображениями I и K вычисляется как

$$PSNR_{дБ} = 10 \lg \frac{MAX_I^2}{MSE} = 20 \lg \frac{MAX_I}{\sqrt{MSE}}, \quad (27)$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2. \quad (28)$$

MSE – среднеквадратичное значение ошибки (Mean Squared Error), MAX_I – максимальное возможное значение яркости пикселя на изображениях I и K размерами $m \times n$.

Например, если в изображении один пиксель кодируется 8 битами, тогда $MAX_I = 255$. Для многоканальных изображений, к примеру RGB (Red Green Blue), MSE рассчитывается как среднее арифметическое среднеквадратических значений ошибки всех трёх цветовых каналов.

Описанные выше метрики позволяют оценить изменение уровня шума на изображении в результате фильтрации, однако они довольно плохо отражают улучшение качества в смысле визуального восприятия человеком. Для этого часто применяется метрика SSIM (Structural Similarity Index) – индекс структурного сходства. Данная метрика вычисляет оценку визуальной схожести двух изображений с помощью учёта их структурной информации.

Для двух изображений I и K SSIM вычисляется по формуле

$$SSIM(I, K) = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)}, \quad (29)$$

где μ_I, μ_K – средние значения на изображениях I и K соответственно σ_I^2, σ_K^2 – дисперсии I и K , σ_{IK} – ковариация I и K , $c_1 = (0.01L)^2$, $c_2 = (0.03L)^2$, L – максимум диапазона значений, принимаемых пикселями (255 для 8-битных изображений). SSIM принимает значения в промежутке от -1 до 1, причём данная мера сходства принимает максимальное значение, равное единице, при совпадении двух изображений [3].

Рассматривая значения метрик SNR, PSNR и SSIM, рассчитанных для изображения до и после фильтрации, можно судить о том, как хорошо тот или иной фильтр справляется с задачей подавления шумов. Сравнивая результаты, полученные для одного и того же изображения, но разными фильтрами, можно сделать вывод, насколько один метод фильтрации эффективнее другого.

1.4. Выводы

В данной главе были рассмотрены теоретические основы фильтрации нормального белого и периодического шумов на изображениях, проведён обзор литературы, посвящённой проблематике исследования.

Изложенные теоретические сведения широко используются инженерами и научными исследователями в области цифровой обработки изображений. Позволяют сформулировать математическую постановку задачи фильтрации нормального белого и периодического шумов, а также описывают способы оценки эффективности фильтров.

Следующая глава посвящена описанию способов оценки параметров шумов, реализаций алгоритмов “жёсткой” и “мягкой” пороговой фильтрации применимых к обработке ДПФ изображения и деталей реализации фильтра Винера для устранения нормального белого и периодического шумов. Также в ней предложен подход масочной обработки цифрового изображения частотными методами для более качественного и эффективного подавления белого гауссова шума.

Глава 2. Особенности и детали реализации исследуемых методов

2.1. Оценка параметров шума на изображении

Для корректной и эффективной работы рассмотренных в прошлой главе фильтров (фильтра Винера, метода оптимальной узкополосной фильтрации) бывает необходимо знать некоторые параметры шума или хотя бы их приближённые оценки. Так для фильтрации нормального белого шума бывает полезно знать оценки его математического ожидания и среднеквадратического отклонения, а для фильтрации периодической помехи нужно определить частоты, составляющих её гармоник.

Часто бывает необходимо оценить параметры нормального белого шума непосредственно по зашумлённому изображению или по набору таких изображений. В этом случае следует выбрать на некотором изображении область, по возможности лишённую характерных черт, чтобы распределение яркости на ней было бы произведено главным образом шумом. По данной области можно вычислить гистограмму яркостей пикселей. Проанализировав полученный результат, можно получить приближённые оценки искомых параметров. Найти точные значения математического ожидания и дисперсии шума в рассматриваемой области не представляется возможным, так как шум наложен на конкретное изображение. Тем не менее, выбирая однородную область без фоновых деталей и полагая шум гауссовым, можно считать, что средняя яркость изображения с шумом приблизительно равна средней яркости этого изображения без шума (считаем, что гауссов шум имеет нулевое среднее). Кроме того, поскольку выбранная область достаточно однородна по яркости, разумно утверждать, что дисперсия прежде всего определяется дисперсией шумового слагаемого [4, стр. 169, 171].

Также параметры шума можно оценивать и по Фурье-образу зашумлённого изображения. Природа спектров реальных изображений чаще всего такова, что с увеличением пространственных частот амплитуда спектральной плотности убывает – это в полной мере справедливо для большинства изображений фотографического характера. Потому с некоторой точностью спектры непрерывных изображений можно считать ограниченными (максимальными) значениями пространственных частот, и по этим ограничениям выбирать периоды дискретизации для преобразования непрерывных изображений в дискретные [2, стр. 128].

Согласно теореме Котельникова для точного восстановления изображения $f(x, y)$, у которого носитель спектра ограничен прямоугольником $\{(u, v) \mid |u| < F_{u \max}, |v| < F_{v \max}\}$, по отсчётам $f_{m,n} = f(m\Delta x, n\Delta y)$ этого изображения, шаги дискретизации должны быть выбраны из условия

$$\Delta x = \frac{1}{2F_1}, \quad \Delta y = \frac{1}{2F_2}, \quad (30)$$

где $F_1 \geq F_{u \max}$, $F_2 \geq F_{v \max}$ [2, стр. 12].

В спектре дискретного изображения, отсчёты которого $F(k, l) = \tilde{F}(\frac{k}{M}, \frac{l}{N})$ могут быть найдены при помощи ДПФ, верхним частотам Ω_H соответствуют индексы k и l , значения которых близки к $M/2$ и $N/2$ соответственно. Эти высокочастотные отсчёты можно считать пренебрежимо малыми (равными нулю), поскольку реально дискретизация непрерывных изображений в пространственной области всегда проводится с некоторым запасом (отсчёты изображения $f_{m,n}$ располагаются более плотно, чем это требует теорема Котельникова). Таким образом, для верхних частот спектра $(k, l) \in \Omega_H$ двумерного сигнала, зашумлённого аддитивным белым шумом, получим:

$$G(k, l) = \underbrace{F(k, l)}_{\approx 0} + N(k, l), \quad \text{или} \quad G(k, l) \approx N(k, l). \quad (31)$$

Для некоррелированного шума $\forall(k, l): M[|N(k, l)|^2] = MN\sigma_\eta^2$ [2, стр. 110]. Поэтому дисперсию такого шума в изображении можно оценить по высокочастотным составляющим Ω_H ДПФ зашумлённого изображения [2, стр. 128]:

$$\sigma_\eta^2 \approx \frac{1}{MN |\Omega_H|} \sum_{(k, l) \in \Omega_H} |G(k, l)|^2, \quad (32)$$

где $|\Omega_H|$ - количество отсчётов ДПФ в высокочастотной области, которые были взяты для нахождения оценки.

Чтобы определить частоты гармоник периодической помехи можно воспользоваться тем, что преобразование Фурье косинусоиды в чистом виде представляет собой пару сопряжённых импульсов, расположенных в центрально-симметричных точках частотной области, которые отвечают частотам косинусоидальной волны. Поэтому если амплитуды косинусоидальных волн в пространственной области достаточно велики, можно ожидать, что в спектре изображения будут видны пары импульсов, по одной для каждой гармоники [1,

стр. 374-375]. Такой результат получается для непрерывных функций и бесконечной длины окна наблюдения. Для цифровых изображений спектр гармонической помехи выглядит как пара сопряжённых *sinc*-функций, этот эффект называется “утечкой спектра” [1, стр. 301]. Пример зашумлённого изображения и его амплитудного спектра приведён на рисунках 4 и 5.



Рис. 4. Изображение с наложенной периодической помехой

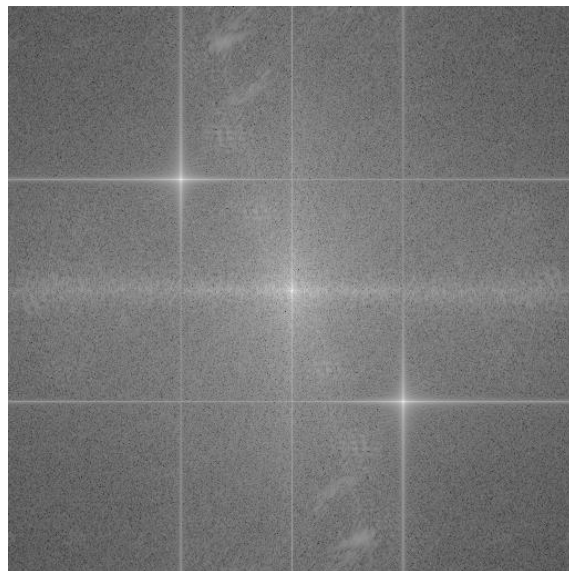


Рис. 5. Амплитудный спектр зашумлённого изображения

На амплитудном спектре зашумлённого изображения отчётливо выделяются отсчёты, соответствующие частотам гармоник шума (рис. 5), их амплитуды значительно больше, чем у соседних коэффициентов. Таким образом, рассмотрев спектр самостоятельно или проанализировав его с помощью специальных программных средств, мы можем определить, какие отсчёты на спектре нужно подавить, чтобы избавиться от шума на изображении.

2.2. Реализация методов “жёсткой” и “мягкой” пороговой фильтрации применимых к обработке ДПФ зашумлённого изображения

Как правило, спектр мощности шумовой составляющей бывает сильно меньше спектра мощности полезного сигнала. А значит к Фурье-образу зашумлённого изображения можно применять пороговые фильтры. Подавив коэффициенты, соответствующие шумовой составляющей, и применив затем обратное ДПФ, можно уменьшить мощность шума на изображении.

Описанные в предыдущей главе методы пороговой фильтрации используются в области дискретного вейвлет-преобразования (ДВП) [6, стр. 157-158]. Отсчёты ДПФ в

отличие от отсчётов ДВП являются комплексными величинами. В связи с этим приведённый выше алгоритм “жёсткой” пороговой фильтрации с функцией (16) можно применять к ДПФ цифрового изображения, однако алгоритм “мягкой” пороговой фильтрации с функцией (17) к ДПФ применять нельзя. Связано это с тем, что функция $sign(x)$ определена только на множестве действительных чисел \mathbb{R} , а отсчёты ДПФ содержат вещественную и мнимую части.

Поэтому для частотной пороговой фильтрации необходимо использовать такие функции, которые подавляли бы амплитуды коэффициентов спектра $F(k, l)$, соответствующих шумовой составляющей, при этом не меняя значения их аргументов $\varphi = arg(F(k, l))$. Если учесть всё это, то можно записать формулы для “жёсткой” и “мягкой” пороговой фильтрации в частотной области.

Таким образом, для “жёсткого” фильтра следует использовать функцию:

$$hard(z, \lambda) = \begin{cases} z, & \text{если } |z| \geq \lambda; \\ 0, & \text{если } |z| < \lambda, \end{cases} \quad (33)$$

где z – коэффициент частотного спектра, λ – заданный порог.

А для “мягкого” фильтра:

$$soft(z, \lambda) = \begin{cases} \frac{z}{|z|} \cdot (|z| - \lambda), & \text{если } |z| \geq \lambda; \\ 0, & \text{если } |z| < \lambda. \end{cases} \quad (34)$$

Спектральная плотность нормального белого шума представляет собой константу, а ДПФ от суммы исходного изображения и шумовой составляющей равно сумме их ДПФ. Поэтому получается, что вклад от спектра мощности белого шума содержится в каждом отсчёте Фурье-образа зашумлённого изображения. “Мягкий” пороговый фильтр как бы “подтягивает” отсчёты, модуль которых больше λ , к нулю, в то время как “жёсткий” не изменяет их. Следовательно, “мягкая” фильтрация уменьшает вклад спектральной мощности шума во всех отсчётах, в отличие от “жёсткой”, которая только обнуляет отсчёты, модули которых меньше заданного порога λ .

Оба фильтра решают задачу подавления нормального белого шума. Результат фильтрации зашумлённого изображения “Barbara” (см. рис. 6) каждым из фильтров можно видеть на рисунках 7 и 8 соответственно.

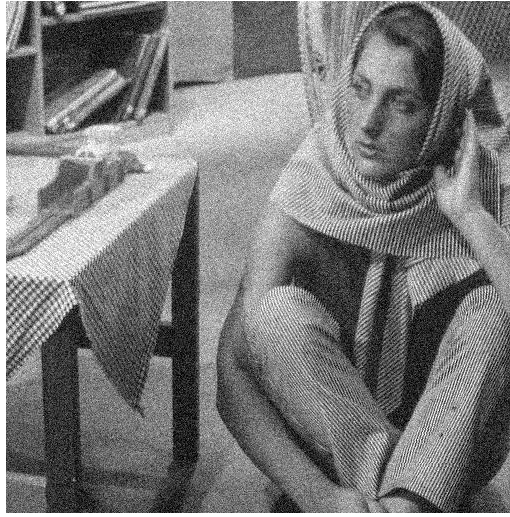


Рис. 6. Зашумлённое изображение “Barbara”



Рис. 7. “Жёсткий” фильтр
($\lambda_{\text{жестк.}} = 0.16$)



Рис. 8. “Мягкий” фильтр
($\lambda_{\text{мягк.}} = 0.076$)

Значения метрик до и после “жёсткой” пороговой фильтрации: SNR (16.2/24.08), SSIM (0.47/0.85); “мягкой” пороговой фильтрации: SNR (16.2/23.81), SSIM (0.47/0.83).

2.3. Реализация фильтра Винера для фильтрации нормального белого шума

Для применения фильтра Винера необходимо знать оценки спектров мощности полезного сигнала и шумовой составляющей. Перепишем формулу частотной характеристики фильтра Винера (20) следующим образом:

$$W(u, v) = \frac{S_g(u, v) - \tilde{S}_\eta(u, v)}{S_g(u, v)}. \quad (35)$$

где $S_g(u, v)$ – спектр мощности зашумлённого изображения, $\tilde{S}_\eta(u, v)$ – оценка спектра мощности шума, и соответственно $S_g(u, v) - \tilde{S}_\eta(u, v) = \tilde{S}_f(u, v)$ – оценка спектра мощности исходного изображения.

Зная, что спектр мощности рассматриваемого шума является константой, можно выразить его через сумму квадратов модулей отсчётов ДПФ шумовой составляющей:

$$S_{\eta}(u, v) = S_{\eta} = \frac{1}{|\Omega|} \sum_{(k, l) \in \Omega} |N(k, l)|^2, \quad (36)$$

Ω – множество отсчётов ДПФ шума, а $|\Omega|$ – их количество.

Используя соотношение (32), можно выразить оценку спектра мощности шума $\tilde{S}_{\eta} \approx S_{\eta}$ через дисперсию или высокочастотные составляющие ДПФ зашумлённого изображения:

$$\tilde{S}_{\eta} = MN\sigma_{\eta}^2 \approx \frac{1}{|\Omega_H|} \sum_{(k, l) \in \Omega_H} |G(k, l)|^2 \quad (37)$$

Подставив найденную оценку \tilde{S}_{η} в выражение (35), мы получим частотную характеристику фильтра Винера. Чтобы получить восстановленное изображение, нужно найти обратное ДПФ от произведения $W(u, v)$ на спектр зашумлённого изображения $G(u, v)$:

$$\hat{f}(x, y) = \mathfrak{F}^{-1}[W(u, v)G(u, v)] \quad (38)$$

2.4. Реализация фильтра Винера для фильтрации периодического шума

Чтобы использовать винеровскую фильтрацию для устранения периодической помехи, также необходимо знать оценку $\tilde{S}_{\eta}(u, v)$ её спектра мощности.

Зная, что ДПФ косинусоиды представляет собой пару сопряжённых *sinc*-функций, мы можем, определив амплитуду A , а также частоты помехи u_0 и v_0 , смоделировать аналогичный двумерный периодический сигнал $r(x, y)$. Его спектр мощности следует использовать как оценку спектра мощности помехи:

$$\tilde{S}_{\eta}(u, v) = S_r(u, v) = |R(u, v)|^2 \quad (39)$$

где $R(u, v) = \mathfrak{F}[r(x, y)]$.

Далее также, как в случае с нормальным шумом, с помощью формулы (35) найдём частотную характеристику фильтра Винера и, используя выражение (38), получим восстановленное изображение $\hat{f}(x, y)$.

2.5. Скользящее окно и весовая маска

Одной из важных особенностей программной реализации методов фильтрации нормального белого шума, описанных в этой главе, является использование скользящего окна. Участок изображения, попавший в окрестность окна, расширяется добавлением заданного числа нулевых отсчётов по краям, затем вычисляется его ДПФ и происходит дальнейшая обработка. Расширение нулями, позволяет получить ДПФ, содержащее большее количество отсчётов, а значит лучше аппроксимирующее непрерывный спектр изображения [8, стр. 67]. Таким образом, частотные методы получают больше информации о спектре, что приводит к увеличению эффективности фильтрации. После обработки участка вычисляется обратное ДПФ, окрестность, содержащая отсчёты, соответствующие обработанным пикселям, используется для формирования отфильтрованного изображения, а окно переходит к следующему участку.

Окно проходит по изображению с некоторым заданным шагом и обрабатывает отсчёты, попавшие в него. Причём шаг подбирается так, чтобы смежные окна немного перекрывались друг с другом.

Это позволяет сделать фильтрацию более аккуратной, локально обрабатывая разные участки изображения, и получать визуально более качественный результат. На спектрах однородных участков вся полезная информация содержится в отсчётах низких частот, а отсчётам высоких частот соответствует только шум. Однако, если работать с ДПФ всего изображения, мы столкнёмся с тем, что высокочастотной области будут также соответствовать контуры и мелкие детали, фильтровать которые не нужно.

Но тогда области, по которым будут перекрываться смежные окна, мы будем фильтровать несколько раз, поэтому их нужно фильтровать “слабее” для каждого отдельного участка, попавшего в окно. Этого можно добиться, используя весовую маску, на которую каждое окно будет умножаться перед фильтрацией.

Маска представляет собой произведение вектора-столбца действительных чисел v заданного размера N на себя самого транспонированного:

$$vv^T = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} (v_1 \ v_2 \ \dots \ v_N) = \begin{pmatrix} v_1 v_1 & \dots & v_1 v_N \\ \vdots & \ddots & \vdots \\ v_N v_1 & \dots & v_N v_N \end{pmatrix}. \quad (40)$$

Например, пусть $v = (0.5, 1, 0.5)^T$, тогда его маска M имеет вид:

$$M = vv^T = \begin{pmatrix} 0.5 \\ 1 \\ 0.5 \end{pmatrix} (0.5, 1, 0.5) = \begin{pmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{pmatrix}. \quad (41)$$

Коэффициенты в окрестности середины вектора принимают значения равные 1, остальные – значения последовательности действительных чисел из отрезка $[0; 1]$, монотонно убывающей к концам вектора. Соседние маски перекрываются по областям монотонного убывания весовых коэффициентов так, чтобы сумма значений перекрывающихся коэффициентов равнялась 1.

В работе использовалась маска, матрица значений которой является произведением векторов vv^T , где v – вектор-столбец длины 40, 8 центральных элементов которого равны 1, а элементы по краям убывают до 0 по закону:

$$f(x_i) = \frac{\cos\left(\frac{\pi i}{15}\right) + 1}{2}, \quad i = 0, 1, \dots, 15. \quad (42)$$

В ходе экспериментов было установлено, что при данных подобранных значениях весов результаты фильтрации всеми фильтрами дают наибольшее увеличение SNR, PSNR и SSIM. Трёхмерное изображение маски представлено на рис. 9.

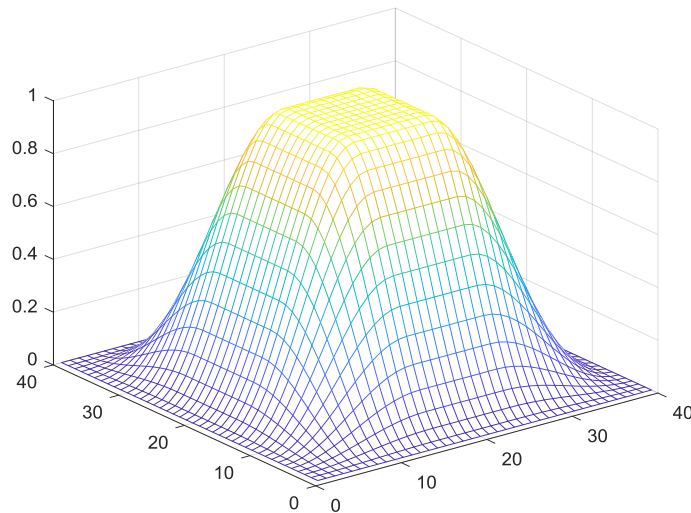


Рис. 9. Трёхмерное изображение маски

По рисункам 10 и 11 можно оценить, как меняется качество фильтрации при использовании окна и без него. На них представлены результаты обработки изображения “Boat”, зашумлённого нормальным белым шумом со среднеквадратическим отклонением $\sigma = 20$ и математическим ожиданием $\mu = 0$, двумя реализациями “жёсткого” порогового фильтра. Одна из них с применением скользящего окна, а другая фильтрует изображение целиком ($\lambda_{жестк.} = 0.6$).

Значения метрик до и после оконной “жёсткой” пороговой фильтрации: SNR (16.78/22.75), SSIM (0.43/0.72); “жёсткой” пороговой фильтрации всего изображения целиком: (SNR 16.78/15.6), SSIM (0.43/0.48). Можно видеть, что SNR во втором случае уменьшается после обработки.

Использование окна и маски позволяет сделать фильтрацию более аккуратной, меньше размывая изображение.



Рис. 10. “Жёстк.” фильтр (с окном)



Рис. 11. “Жёстк.” фильтр (целиком)

2.6. Выводы

В данной главе были описаны способы оценки параметров нормального белого и периодического шумов. На основе изученной теории предложены реализации алгоритмов “жёсткой” и “мягкой” пороговой фильтрации белого гауссового шума применимые к обработке ДПФ зашумлённого изображения. Также рассмотрены две реализации фильтра Винера для фильтрации нормального белого и периодического шумов.

На основе этой информации была разработана программа на языке MATLAB, моделирующая описанные виды шумов, а также реализованы методы их фильтрации.

Следующая глава содержит результаты тестирования всех описанных выше фильтров, а также анализ их эффективности, основанный на сравнении значений метрик SNR, PSNR и SSIM до и после фильтрации.

Глава 3. Тестирование фильтров, анализ результатов и выводы

3.1. Результаты фильтрации тестовых изображений

Все фильтры были протестированы на стандартных чёрно-белых изображениях (512×512 пикселей, 8 бит на пиксель), используемых для проверки качества работы алгоритмов цифровой обработки: “Barbara”, “Boat”, “Lena” и “Goldhill”. По результатам экспериментов построены таблицы, содержащие значения метрик до и после применения фильтров, на основании которых проводилась оценка эффективности.

Для начала рассмотрим фильтрацию нормального белого шума с математическим ожиданием $\mu = 0$ и среднеквадратичным отклонением $\sigma = 20$. Размеры окон для пространственных фильтров – 3×3, для фильтра Гаусса $\sigma_D = 0.9$, для билатерального фильтра $\sigma_d = 1.2$, $\sigma_r = 80$, для “жёсткого” порогового фильтра $\lambda_{жестк.} = 0.16$, для “мягкого” порогового фильтра $\lambda_{мягк.} = 0.076$. Все частотные фильтры использовали оконную фильтрацию с весовой маской, представленной в главе 2 (размерами 40×40), перед вычислением ДПФ обрабатываемые участки изображения дополнялись нулевыми отсчётами до размера 80×80. Параметры фильтров были подобраны так, чтобы в результате фильтрации получалось наибольшее увеличение метрик SNR, PSNR и SSIM. В качестве оценки $\tilde{S}_\eta(u, v)$ в ЧХ фильтра Винера использовался средний квадрат модуля ДПФ чистой шумовой составляющей, вычисленный по формуле (36).

Таким образом, были получены 7 таблиц.

Таблица № 1. Значения метрик до и после фильтрации изображения среднеарифметическим фильтром

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	16.2/18.66	16.78/22.15	16.79/23.48	15.75/22.16
PSNR, дБ до/после	22.12/24.58	22.12/27.5	22.12/28.82	22.12/28.53
SSIM до/после	0.47/0.65	0.43/0.7	0.39/0.75	0.4/0.72

Таблица № 2. Значения метрик до и после фильтрации изображения фильтром срединной точки

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	16.2/17.63	16.78/20.18	16.79/21.03	15.75/20.36
PSNR, дБ до/после	22.12/23.55	22.12/25.52	22.12/26.37	22.12/26.73
SSIM до/после	0.47/0.56	0.43/0.6	0.39/0.65	0.4/0.62

Таблица № 3 Значения метрик до и после фильтрации изображения фильтром Гаусса

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	16.2/19.72	16.78/22.75	16.79/23.75	15.75/22.54
PSNR, дБ до/после	22.12/25.64	22.12/28.1	22.12/29.09	22.12/28.91
SSIM до/после	0.47/0.69	0.43/0.71	0.39/0.74	0.4/0.73

Таблица № 4. Значения метрик до и после фильтрации изображения билатеральным фильтром

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	16.2/20.25	16.78/23.39	16.79/24.45	15.75/23.12
PSNR, дБ до/после	22.12/26.17	22.12/28.74	22.12/29.79	22.12/29.49
SSIM до/после	0.47/0.74	0.43/0.76	0.39/0.81	0.4/0.77

Таблица № 5. Значения метрик до и после фильтрации изображения “жёстким” пороговым фильтром

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	16.2/24.08	16.78/23.32	16.79/24.95	15.75/22.78
PSNR, дБ до/после	22.12/30	22.12/28.66	22.12/30.29	22.12/29.15
SSIM до/после	0.47/0.85	0.43/0.75	0.39/0.85	0.4/0.75

Таблица № 6. Значения метрик до и после фильтрации изображения “мягким” пороговым фильтром

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	16.2/23.81	16.78/23.36	16.79/24.6	15.75/22.91
PSNR, дБ до/после	22.12/29.73	22.12/28.7	22.12/29.93	22.12/29.27
SSIM до/после	0.47/0.83	0.43/0.75	0.39/0.81	0.4/0.75

Таблица № 7. Значения метрик до и после фильтрации изображения фильтром Винера

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	16.2/24.09	16.78/23.27	16.79/24.9	15.75/22.82
PSNR, дБ до/после	22.12/30.01	22.12/28.62	22.12/30.24	22.12/29.19
SSIM до/после	0.47/0.86	0.43/0.76	0.39/0.86	0.4/0.76

Наблюдаемое увеличение SNR и PSNR после обработки частотными методами в большинстве случаев больше, чем у пространственных фильтров (разница составляет от 0.5 до 6.45 дБ). Некоторые изображения после обработки частотными фильтрами показывают увеличение SSIM в 1.05-1.54 раз больше, по сравнению с результатами пространственных методов. Наилучшие результаты среди пространственных фильтров демонстрирует билатеральный. В отдельных случаях его применение позволяет добиться большего увеличения метрик, чем у частотных методов. В то же время фильтры, основывающиеся на анализе и изменении ДПФ, а также билатеральный фильтр, использующий информацию о расстоянии между обрабатываемыми пикселями и разнице их яркостей, требуют больше вычислительных операций. Наибольший рост SNR среди частотных методов демонстрируют “жесткий” пороговый фильтр и фильтр Винера. Самое эффективное подавление шума в смысле увеличения SNR и PSNR можно видеть на примере “жесткой” пороговой фильтрации изображения “Lena”, SNR (16.79/24.95); SSIM (0.39/0.85) (рис. 12-14).



*Рис. 12. Исходное
незашумлённое
изображение*

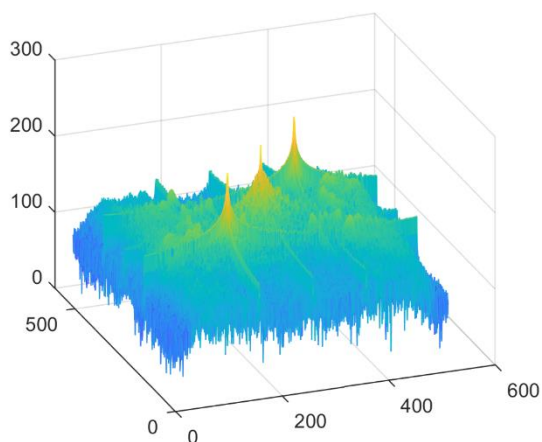


*Рис. 13. Зашумлённое
изображение*

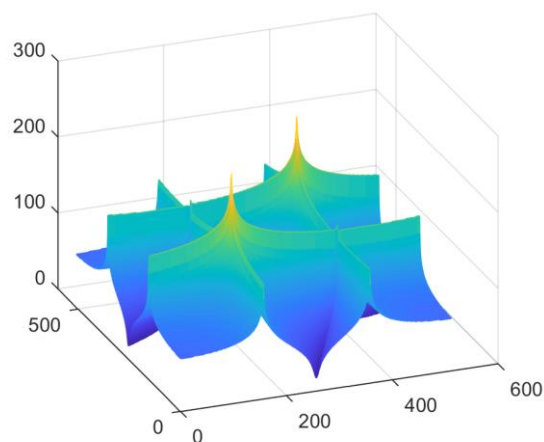


*Рис. 14. Обработанное
изображение*

Теперь рассмотрим фильтрацию периодического шума с амплитудой $A = 90$ и частотами $u_0 = 100.4$, $v_0 = 100.2$. При тестировании метода оптимальной узкополосной фильтрации для выделения шумовой составляющей использовались частотный фильтр Гаусса ($\sigma_G = 22$) и частотный фильтр Баттерворта ($\sigma_B = 25$, $n = 11$). В результате экспериментов было установлено, что выбранные значения параметров обеспечивают наибольшее увеличение метрик после фильтрации. В качестве оценки спектра мощности помехи в ЧХ фильтра Винера использовался спектр мощности смоделированного с помощью формулы (4) двумерного периодического сигнала с заданными значениями частот $\hat{u}_0 = u_0$ и $\hat{v}_0 = v_0$ и амплитудой $\hat{A} = A$. На рисунках 15 и 16 представлены 3-D графики спектра мощности изображения “Barbara”, зашумлённого периодической помехой, и спектра мощности смоделированной помехи соответственно.



*Рис. 15. Спектр мощности
зашумлённого изображения*



*Рис. 16. Спектр мощности
смоделированной помехи*

По результатам экспериментов были построены 3 таблицы.

Таблица № 8. Значения метрик до и после фильтрации изображения методом оптимальной узкополосной фильтрации (фильтр Гаусса)

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	6.14/26.78	6.71/31.02	6.72/27.81	5.69/26.59
PSNR, дБ до/после	12.06/32.7	12.06/36.36	12.06/33.15	12.06/32.95
SSIM до/после	0.13/0.96	0.1/0.97	0.1/0.97	0.08/0.96

Таблица № 9. Значения метрик до и после фильтрации изображения методом оптимальной узкополосной фильтрации (фильтр Баттерворта)

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	6.14/26.43	6.71/30.62	6.72/27.61	5.69/26.39
PSNR, дБ до/после	12.06/32.35	12.06/35.96	12.06/32.94	12.06/32.75
SSIM до/после	0.13/0.95	0.1/0.96	0.1/0.95	0.08/0.95

Таблица № 10. Значения метрик до и после фильтрации изображения фильтром Винера (фильтрация периодической помехи)

Изображение	Barbara.png	Boat.png	Lena.png	Goldhill.png
SNR, дБ до/после	6.14/33.23	6.71/36.37	6.72/ 35.61	5.69/36.04
PSNR, дБ до/после	12.06/39.15	12.06/41.71	12.06/40.95	12.06/42.41
SSIM до/после	0.13/0.97	0.1/0.98	0.1/0.98	0.08/0.98

Видно, что при использовании винеровской фильтрации наблюдается более эффективное подавление периодической помехи, чем при применении обоих вариантов метода оптимальной узкополосной фильтрации. Для разных изображений фильтр Винера демонстрирует увеличение SNR и PSNR примерно на 7-10 дБ больше. Увеличение SSIM при обработке фильтром Винера всех тестовых изображений больше, чем при использовании метода оптимальной узкополосной фильтрации с применением фильтров Баттерворта и Гаусса для выделения шумовой составляющей. При сравнении двух реализаций метода оптимальной узкополосной фильтрации видно,

что вариант с фильтром Гаусса является более эффективным, однако разница в росте SNR не превышает 0.4 дБ, разница в росте SSIM не превышает 0.02. Пример винеровской фильтрации с наибольшим подавлением периодической помехи приведён на рисунках 17-19, SNR (5.69/36.04); SSIM (0.08/0.98).



Рис. 17. Исходное незашумлённое изображение



*Рис. 18. Изображение “Goldhill”,
зашумлённое периодическим шумом*



*Рис. 19. Отфильтрованное
изображение “Goldhill”*

Примеры обработанных изображений, соответствующих данным в таблицах, а также код программных реализаций фильтра Винера, методов “жёсткой” и “мягкой” пороговой фильтрации и метода оптимальной узкополосной фильтрации на языке MATLAB приведены в приложениях.

3.2. Вычислительная сложность исследуемых методов фильтрации

Определим вычислительные сложности методов фильтрации нормального белого шума, продемонстрировавших наибольшее увеличение метрик, – “жёсткого” порогового фильтра и фильтра Винера.

Оба метода проходят по изображению скользящим окном. Число операций, выполняемых каждым фильтром, равно произведению числа операций, требующихся для фильтрации одного фрагмента изображения, попавшего в окно, и количества обрабатываемых фрагментов. На рисунках 20 и 21 соответственно представлены схема маски и схема покрытия окнами размера $m \times m$ с областью единичных значений $l \times l$ изображения размерами $N \times N$. Окно переходит к следующему фрагменту с шагом равным $(m + l)/2$ пикселей. Дойдя до правого конца, окно с этим же шагом сдвигается вниз и возвращается в левый конец изображения – в начало следующего ряда – и так продолжает обработку, пока не пройдёт по всему изображению. При движении окна вдоль одного ряда фильтр обработает $n_1 = \frac{N+(m-l)/2}{(m+l)/2} = \frac{2N+m-l}{m+l}$ фрагментов, а общее их число на всём изображении равно $n = n_1^2 = \frac{(2N+m-l)^2}{(m+l)^2}$.

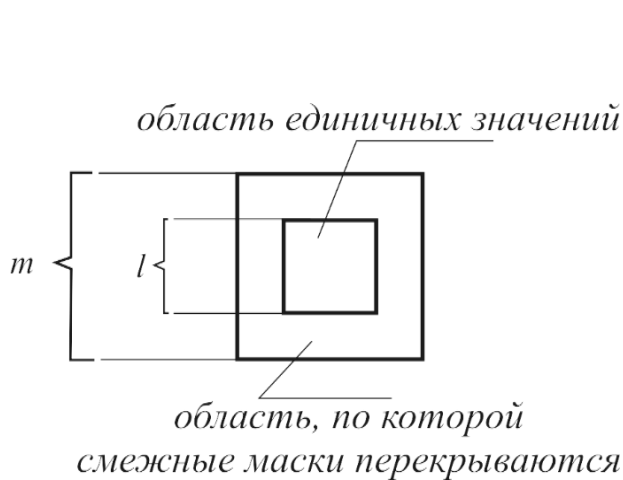


Рис. 20. Схема маски

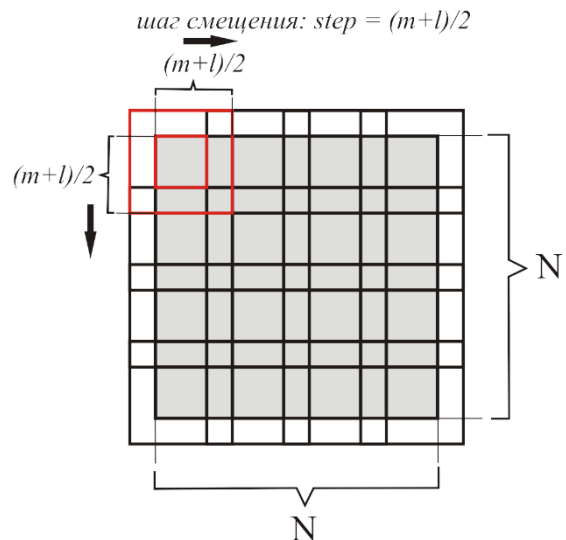


Рис. 21. Схема покрытия изображения окнами

Вычисление двумерного быстрого преобразования Фурье (БПФ) требует $N^2 \log_2(N)$ комплексных умножений и $2N^2 \log_2(N)$ комплексных сложений [9]. Перед вычислением БПФ рассматриваемые методы дополняют обрабатываемый фрагмент изображения нулевыми отсчётами. Пусть S – коэффициент, отвечающий за то, во сколько раз увеличиваются размеры окна после дополнения нулевыми отсчётами.

Тогда на каждой итерации мы находим БПФ от фрагмента размерами $(mS) \times (mS)$. Одной операции комплексного умножения соответствуют 4 операции вещественного умножения, а одной операции комплексного сложения соответствуют 2 операции вещественного сложения. При обработке каждого фрагмента вычисляется 2 БПФ: прямое и обратное. Тогда всего получим $16(mS)^2 \log_2(mS)$ операций.

В “жестком” пороговом фильтре помимо вычисления БПФ на каждый фрагмент приходится m^2 вещественных умножений отсчетов фрагмента на веса маски, $(mS)^2$ комплексных умножений, $(mS)^2$ сравнений с порогом λ , k присваиваний нуля ($k \leq (mS)^2$, верхней оценкой будет случай, когда $k = (mS)^2$) и $m^2 n$ сложений для формирования обработанного изображения. Для обоих фильтров заданы следующие значения параметров: $S = 2, m = 40, l = 8$. Сложив общее число операций и поделив на количество пикселей (512×512), получим не более 1272 операций на пиксель. Тогда для обработки изображения размерами $N \times N$ в среднем необходимо выполнить $1272N^2$ операций.

Реализация фильтра Винера помимо вычисления БПФ для каждого фрагмента выполняет m^2 вещественных умножений, $2(mS)^2$ комплексных умножений, $(mS)^2$ вещественных вычитаний, столько же сравнений с нулём, k присваиваний нуля ($k \leq (mS)^2$, положим $k = mS$), $(mS)^2$ вещественных делений и $m^2 n$ сложений. В среднем на один пиксель приходится не более 1342 операций. Для обработки изображения размерами $N \times N$ в среднем необходимо выполнить $1342N^2$ операций. Асимптотическая сложность обоих методов: $O(N^2)$.

3.3. Выводы

Проанализировав полученные данные, можно сделать вывод, что при фильтрации нормального белого шума частотные фильтры позволяют получить более качественный результат в смысле большего увеличения SNR, PSNR и SSIM, чем пространственные. Установлено, что среди частотных методов наибольший рост метрик демонстрируют “жесткая” пороговая фильтрация (PSNR, SNR) и фильтр Винера (SSIM). Их вычислительная сложность является квадратичной $O(N^2)$.

В то же время пространственные фильтры гораздо проще в реализации и не требуют вычисления и обработки ДПФ изображений. Среди них самое большое увеличение метрик наблюдается при применении билатерального фильтра, учитывающего расстояния между пикселями, а также разницу их яркостей. В некоторых отдельных случаях применение билатеральной фильтрации позволяет добиться большего увеличения метрик, чем при обработке частотными методами.

По результатам экспериментов фильтрации периодического шума видно, что фильтр Винера значительно лучше справляется с задачей, чем метод оптимальной узкополосной фильтрации. Наблюдаемое увеличение SNR и PSNR почти на 7-10 дБ больше, чем у обеих модификаций метода оптимальной узкополосной фильтрации. Однако следует понимать, что здесь большую роль играет то, как точно будут оценены частоты помехи, с которой борется фильтр. Для наиболее эффективной фильтрации необходимо использовать в качестве оценки $\tilde{S}_\eta(u, v)$ спектр мощности смоделированного двумерного периодического сигнала, частоты и амплитуда которого как можно лучше соответствуют частотам и амплитуде помехи. Если сравнивать две реализации второго метода, то вариант с фильтром Гаусса показывает более хорошие результаты, хотя разница в увеличении метрик невелика.

Заключение

В данной работе были исследованы пространственные и частотные методы фильтрации шумов на изображениях. Рассмотрены аддитивные модели двух часто встречающихся в природе шумов, возникающих как результат взаимодействия множества факторов, связанных с процессом получения и передачи изображений, а также внутренним устройством электронных средств, с помощью которых формируются цифровые изображения. Были предложены реализации алгоритмов пороговой фильтрации применимые к обработке ДПФ для подавления нормального белого шума. Описан подход масочной частотной фильтрации белого гауссового шума на изображениях.

Установлено, что с задачей подавления нормального белого шума эффективней, с точки зрения большего увеличения метрик SNR, PSNR и SSIM, справляются частотные методы фильтрации. При устранении периодической помехи самой эффективной оказалась винеровская фильтрация. Также установлено, что фильтр Винера является универсальным средством подавления шумов в том смысле, что подходит для борьбы с шумами различной природы.

Однако, если использовать полученные сведения в разработке программ и электронных средств, решающих задачу фильтрации, следует помнить, что методы, основывающиеся на анализе и изменении спектра изображения, бывают затратными по времени и ресурсам. Их техническая реализация может оказаться достаточной трудоёмкой. Поэтому нельзя сказать, что применение пространственных фильтров однозначно хуже. Среднеарифметический фильтр, фильтр Гаусса и фильтр срединной точки, уступая в качестве работы, обладают гораздо меньшей вычислительной сложностью. Более эффективные и затратные по ресурсам фильтры можно использовать в программах цифровой обработки, например таких как Photoshop, а более быстрые методы подойдут там, где важна скорость принятия решений, на основе информации, получаемой с изображений.

В любом случае выбор метода фильтрации всегда должен исходить из имеющихся вычислительных мощностей и особенностей конкретной задачи.

Список источников

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2012. – 1104 с.
2. С.В. Умняшкин, В.В. Лесин. Основы цифровой обработки изображений: учебное пособие. – М.: МИЭТ, 2016. – 200 с.
3. Ашрапов И. "PSNR и SSIM или как работать с изображениями под C" – URL: <https://habr.com/ru/post/126848/>
4. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. – М.: Техносфера, 2006. – 616 с.
5. M. Elad, "On the bilateral filter and ways to improve it", // IEEE Transactions on Image Processing, vol.11, 1141-1151, 2002.
6. Воскобойников Ю. Е., Гочаков А. В., Колкер А. Б. Фильтрации сигналов и изображений: Фурье и вейвлет алгоритмы (с примерами в Mathcad): монография. – Новосибирск: НГАСУ (Сибстрин), 2010. – 188 с.
7. Умняшкин С. В. Основы теории цифровой обработки сигналов: Учебное пособие. Издание пятое, исправленное и дополненное. – М.: Техносфера, 2019. – 550 с.
8. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. – М.: Издательство «МИР», 1978, – 848 с.
9. Университет ИТМО Учебное пособие по курсу "Компьютерная обработка изображений" Конспект лекций. – URL: http://dipl.spb.ru/el_books/image_processing/2_04.html

Приложения

Приложение № 1. Среднеарифметическая фильтрация нормального белого шума ($\mu = 0, \sigma = 20$).



Рис. 22. Изображение “Boat”
(зашумлённое)



Рис. 23. Изображение “Boat”
(отфильтрованное)

Приложение № 2. Фильтрация нормального белого шума ($\mu = 0, \sigma = 20$) фильтром срединной точки.

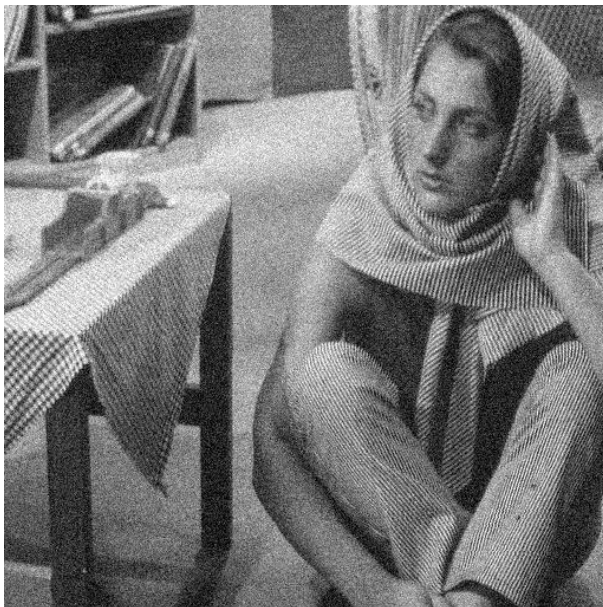


Рис. 24. Изображение “Barbara”
(зашумлённое)

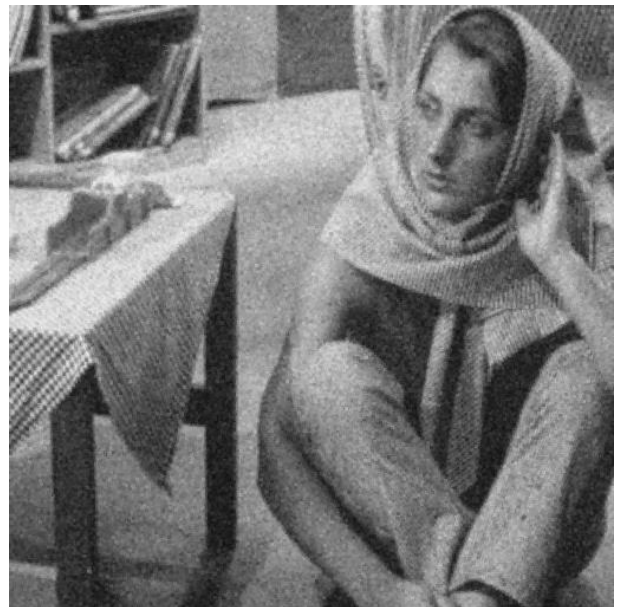


Рис. 25. Изображение “Barbara”
(отфильтрованное)

Приложение № 3. Фильтрация нормального белого шума ($\mu = 0, \sigma = 20$) фильтром Гаусса.



Рис. 26. Изображение “Goldhill”
(зашумлённое)



Рис. 27. Изображение “Goldhill”
(отфильтрованное)

Приложение № 4. Фильтрация нормального белого шума ($\mu = 0, \sigma = 20$) билатеральным фильтром.



Рис. 28. Изображение “Lena”
(зашумлённое)



Рис. 29. Изображение “Lena”
(отфильтрованное)

Приложение № 5. Фильтрация нормального белого шума ($\mu = 0, \sigma = 20$) “жестким” пороговым фильтром.

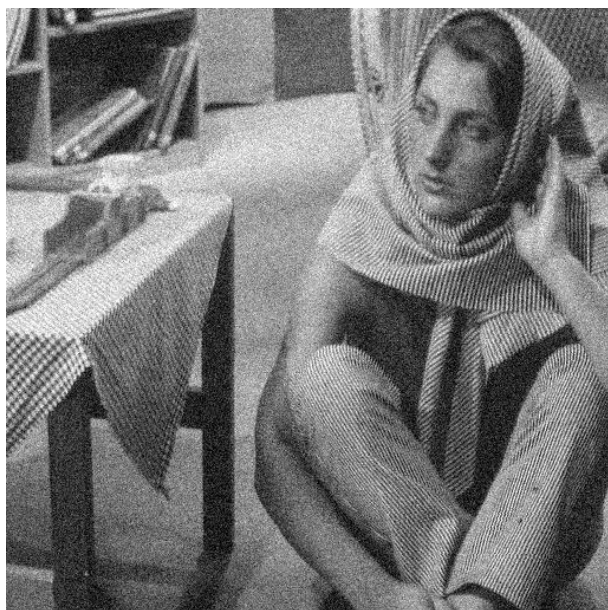


*Рис. 30. Изображение “Boat”
(зашумлённое)*



*Рис. 31. Изображение “Boat”
(отфильтрованное)*

Приложение № 6. Фильтрация нормального белого шума ($\mu = 0, \sigma = 20$) “мягким” пороговым фильтром.



*Рис. 32. Изображение “Barbara”
(зашумлённое)*



*Рис. 33. Изображение “Barbara”
(отфильтрованное)*

Приложение № 7. Фильтрация нормального белого шума ($\mu = 0, \sigma = 20$) фильтром Винера.



*Рис. 34. Изображение “Lena”
(зашумлённое)*



*Рис. 35. Изображение “Lena”
(отфильтрованное)*

Приложение № 8. Фильтрация периодической помехи ($A = 90, u_0 = 100.4, v_0 = 100.2$) методом оптимальной узкополосной фильтрации (для выделения шумовой составляющей использовался фильтр Баттерворта).



*Рис. 36. Изображение “Lena”,
зашумлённое периодическим шумом*



*Рис. 37. Изображение “Lena”
(отфильтрованное)*

Приложение № 9. Фильтрация периодической помехи ($A = 90$, $u_0 = 100.4$, $v_0 = 100.2$) методом оптимальной узкополосной фильтрации (для выделения шумовой составляющей использовался фильтр Гаусса).



*Рис. 38. Изображение "Barbara",
зашумлённое периодическим шумом*



*Рис. 39. Изображение "Barbara"
(отфильтрованное)*

Приложение № 10. Фильтрация периодической помехи ($A = 90$, $u_0 = 100.4$, $v_0 = 100.2$) фильтром Винера.



*Рис. 40. Изображение "Boat",
зашумлённое периодическим шумом*



*Рис. 41. Изображение "Boat"
(отфильтрованное)*

Приложение № 11. Программная реализация метода “жёсткой” пороговой фильтрации (код на языке Matlab).

```
function [im_res] =  
Hard_Filt(im_frame,mask,vect_down_len,k_m,lamb)  
i1 = 1 + vect_down_len;  
i2 = size(im_frame,1) - vect_down_len;  
j1 = 1 + vect_down_len;  
j2 = size(im_frame,2) - vect_down_len;  
im_n = im_frame(i1:i2,j1:j2);  
im_res = zeros(size(im_frame));  
step = size(mask,1) - vect_down_len;  
lamb = lamb*(k_m*size(mask,1))^2;  
lamb2 = lamb*lamb;  
for i = 0:step:size(im_n,1)  
    for j = 0:step:size(im_n,2)  
        i1 = i + 1;  
        i2 = i + size(mask,1);  
        j1 = j + 1;  
        j2 = j + size(mask,2);  
        im_window = double(im_frame(i1:i2,j1:j2)) .* mask;  
        im_temp = zeros(k_m.*size(mask));  
        i1 = (size(im_temp,1) - size(im_window,1))/2;  
        i2 = (size(im_temp,1) + size(im_window,1))/2;  
        j1 = (size(im_temp,2) - size(im_window,2))/2;  
        j2 = (size(im_temp,2) + size(im_window,2))/2;  
        im_temp(i1 + 1:i2,j1 + 1:j2) = im_window;  
        sp_window = fftshift(fft2(im_temp));  
        for k = 1:size(sp_window,1)  
            mas =  
find(sp_window(k,:).*conj(sp_window(k,:))<lamb2);  
            sp_window(k,mas(:)) = 0;  
        end  
        im_temp = ifft2(ifftshift(sp_window));  
        im_window = im_temp(i1 + 1:i2,j1 + 1:j2);  
        i1 = i + 1;  
        i2 = i + size(mask,1);  
        j1 = j + 1;  
        j2 = j + size(mask,2);  
        im_res(i1:i2,j1:j2) = im_res(i1:i2,j1:j2) + im_window;  
    end  
end  
end
```

Приложение № 12. Программная реализация метода “мягкой” пороговой фильтрации (код на языке Matlab).

```
function [im_res] =  
Soft_Filt(im_frame,mask,vect_down_len,k_m,lamb)  
i1 = 1 + vect_down_len;  
i2 = size(im_frame,1) - vect_down_len;  
j1 = 1 + vect_down_len;  
j2 = size(im_frame,2) - vect_down_len;  
im_n = im_frame(i1:i2,j1:j2);  
im_res = zeros(size(im_frame));  
step = size(mask,1) - vect_down_len;  
lamb = lamb*(k_m*size(mask,1))^2;  
for i = 0:step:size(im_n,1)  
    for j = 0:step:size(im_n,2)  
        i1 = i + 1;  
        i2 = i + size(mask,1);  
        j1 = j + 1;  
        j2 = j + size(mask,2);  
        im_window = double(im_frame(i1:i2,j1:j2)) .* mask;  
        im_temp = zeros(k_m.*size(mask));  
        i1 = (size(im_temp,1) - size(im_window,1))/2;  
        i2 = (size(im_temp,1) + size(im_window,1))/2;  
        j1 = (size(im_temp,2) - size(im_window,2))/2;  
        j2 = (size(im_temp,2) + size(im_window,2))/2;  
        im_temp(i1 + 1:i2,j1 + 1:j2) = im_window;  
        sp_window = fftshift(fft2(im_temp));  
        for k = 1:size(sp_window,1)  
            mas1 = find(abs(sp_window(k,:)) < lamb);  
            mas2 = find(abs(sp_window(k,:)) > lamb);  
            sp_window(k,mas1(:)) = 0;  
            for l = 1:length(mas2)  
                z = sp_window(k,mas2(l));  
                koeff = (abs(z) - lamb)/(abs(z));  
                sp_window(k,mas2(l)) = z.*koeff;  
            end  
        end  
        im_temp = ifft2(ifftshift(sp_window));  
        im_window = im_temp(i1 + 1:i2,j1 + 1:j2);  
        i1 = i + 1;  
        i2 = i + size(mask,1);  
        j1 = j + 1;  
        j2 = j + size(mask,2);  
        im_res(i1:i2,j1:j2) = im_res(i1:i2,j1:j2) + im_window;  
    end  
end  
end  
end
```


Приложение № 13. Программная реализация винеровской фильтрации нормального белого шума (код на языке Matlab).

```
function im_res =  
Filtr_Winn(im_frame,noise,mask,vect_down_len,k_m)  
N = size(im_frame,1) - 2*vect_down_len;  
M = size(im_frame,2) - 2*vect_down_len;  
sp_n = fftshift(fft2(noise));  
Sn = size(mask,1)*size(mask,2)*var(sp_n(:))/(M*N);  
im_res = zeros(size(im_frame));  
step = size(mask,1) - vect_down_len;  
for i = 0:step:N  
    for j = 0:step:M  
        i1 = i + 1;  
        i2 = i + size(mask,1);  
        j1 = j + 1;  
        j2 = j + size(mask,2);  
        im_window = double(im_frame(i1:i2,j1:j2)) .* mask;  
        im_temp = zeros(k_m.*size(mask));  
        i1 = (size(im_temp,1) - size(im_window,1))/2;  
        i2 = (size(im_temp,1) + size(im_window,1))/2;  
        j1 = (size(im_temp,2) - size(im_window,2))/2;  
        j2 = (size(im_temp,2) + size(im_window,2))/2;  
        im_temp(i1 + 1:i2,j1 + 1:j2) = im_window;  
        sp_window = fftshift(fft2(im_temp));  
        Sg = sp_window .* conj(sp_window);  
        Sf = Sg - Sn;  
        for k = 1:size(Sf,1)  
            mas = find(Sf(k,:) < 0);  
            for l = 1:length(mas)  
                Sf(k,mas(l)) = 0;  
            end  
        end  
        Filt_winn = Sf ./ Sg;  
        mas = find(Sg == 0);  
        Filt_winn(mas) = 0;  
        Filt_winn(floor(size(sp_window,1)/2) +  
1,floor(size(sp_window,2)/2) + 1) = 1;  
        sp_window = Filt_winn .* sp_window;  
        im_temp = ifft2(ifftshift(sp_window));  
        im_window = im_temp(i1 + 1:i2,j1 + 1:j2);  
        i1 = i + 1;  
        i2 = i + size(mask,1);  
        j1 = j + 1;  
        j2 = j + size(mask,2);  
        im_res(i1:i2,j1:j2) = im_res(i1:i2,j1:j2) + im_window;  
    end  
end  
end  
end
```

Приложение № 14. Программная реализация метода оптимальной узкополосной фильтрации(код на языке Matlab).

```
function im_res = Opt_filtr(im_n,noise,step)
im_res = zeros(size(im_n));
for i = 0:step:size(im_n,1) - step
    for j = 0:step:size(im_n,2) - step
        i1 = i + 1;
        i2 = i + step;
        j1 = j + 1;
        j2 = j + step;
        im_window = double(im_n(i1:i2,j1:j2));
        n_window = double(noise(i1:i2,j1:j2));
        gn_mean = mean(mean(im_window.*n_window));
        g_mean = mean(mean(im_window));
        n_mean = mean(mean(n_window));
        nn_mean = mean(mean(n_window.*n_window));
        w = (gn_mean - g_mean.*n_mean)./(nn_mean - n_mean.^2);
        im_window = im_window - w.*n_window;
        im_res(i1:i2,j1:j2) = im_res(i1:i2,j1:j2) + im_window;
    end
end
end
```

Приложение № 15. Программная реализация винеровской фильтрации периодической помехи (код на языке Matlab).

```
function im_res = Filtr_Winn_garm(im_n,R)
spectr = fftshift(fft2(im_n));
Sg = spectr .* conj(spectr);
Sf = Sg - R.*conj(R);
for k = 1:size(Sf,1)
    mas = find(Sf(k,:) < 0);
    for l = 1:length(mas)
        Sf(k,mas(l)) = 0;
    end
end
Filt_winn = Sf ./ Sg;
Filt_winn(size(spectr,1)/2 + 1,size(spectr,2)/2 + 1) = 1;
spectr = Filt_winn .* spectr;
im_res = ifft2(ifftshift(spectr));
end
```