

НАУЧНО ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ СИСТЕМ УПРАВЛЕНИЯ И РОБОТОТЕХНИКИ

Отчет по лабораторной работе №6
Обработка изображений

Выполнил студент группы Р3380
Преподаватели

Мовчан И.Е.
Пашенко А.В., Перегудин А.А.

Санкт-Петербург
2025

Содержание

1	Фильтрация изображений	2
2	Исследование свёртки	6
2.1	Размытие по Гауссу	6
2.2	Блоchное размытие	14
2.3	Увеличение резкости	24
2.4	Выделение краёв	27
2.5	Своё ядро	29
3	Общие выводы	32
4	Приложение	33

1 Фильтрация изображений

Выберем одно из предложенных изображений с ярко выраженным периодическими шумами поверх него. Используем функцию **imread** и загрузим его в среду matlab для дальнейшего исследования, само изображение представлено ниже.



Рис. 1: Исходное изображение

Можно видеть, что тут явно не помешала бы фильтрация - повсюду гармоники, которые явно не к месту. Попробуем исправить ситуацию - выполним нормализацию изображения, поделив массив значений на 255 - максимальное значение яркости цвета, рассчитаем двумерное дискретное преобразование Фурье с помощью функции **fft2** (для удобства анализа также выполним центрирование за счёт **fftshift**), разделим результат на модуль и фазу, а также выполним

логарифмирование амплитуд, чтобы «уравнять» вклад малозаметных гармоник и ярко выраженного центра (дабы избежать логарифма нуля, к аргументу была добавлена 1), аналогично нормализуем полученный массив для корректного отображения данных.

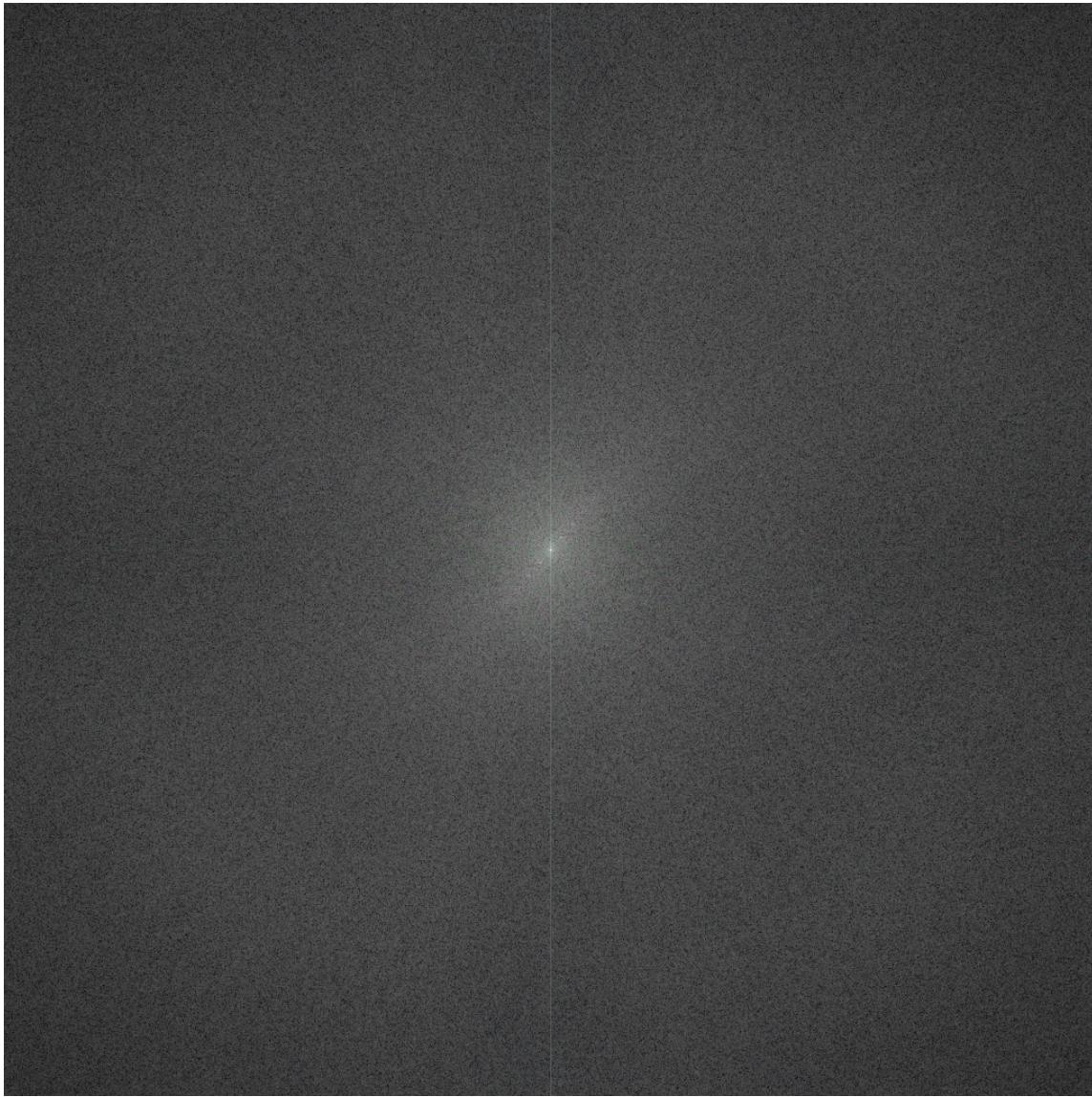


Рис. 2: Нормализованный логарифм модуля спектра

Найденный образ Фурье позволяет выделить те самые лишние гармоники. На рисунках 2 и 3 можно наблюдать выраженные симметричные белые точки вдали от центра, которые соответствуют шу-

мам изображения, - это проявление разложения синусоид, преобразовывающихся в дельта-функции. Эти пики мы и хотим подавить, уменьшив их компоненту до 0. Результаты ручной фильтрации представлены на рисунке 4 - закрашиваются в чёрный цвет те области, в которых наблюдались гармоники (обрабатываются не отдельные точки, так как преобразование Фурье компьютерными средствами приблизительно, и дельта-функции при симуляции проявляются не в конкретных местах, а интервала вокруг них). При восстановлении чёрный цвет превратится в нули на необходимых амплитудных отрезках, остальное же останется нетронутым - этого мы и хотим.

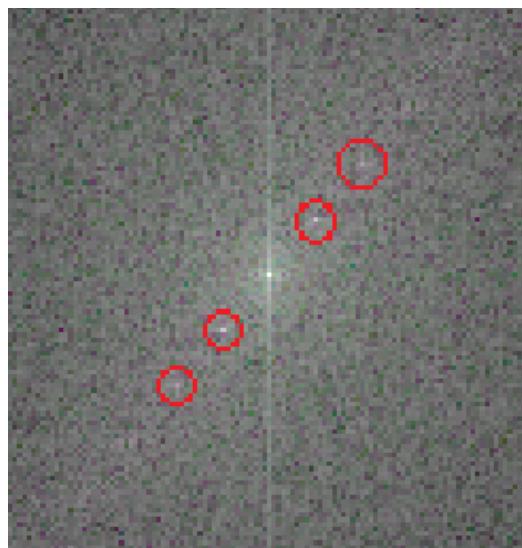


Рис. 3: Обозначение шумов на нормализованном логарифмфе модуля спектра

Теперь выполним обратные описанным выше шаги, то есть умножим на сохраненное максимальное значение логарифма, экспоненцируем, вычтем единицу, возьмём обратное преобразование Фурье и найдём отредактированное изображение через полученные модули и фазы оригинальной картинки. Результаты представлены на рисунке 5а, сравнение с исходными данными - на рисунке 5. На изображении видно, что выраженные шумы были успешно удалены, а общая структура и формы объектов осталась близка к оригинальным.

Таким образом, преобразование Фурье позволило выявить периодические компоненты изображения в частотной области, а также от-

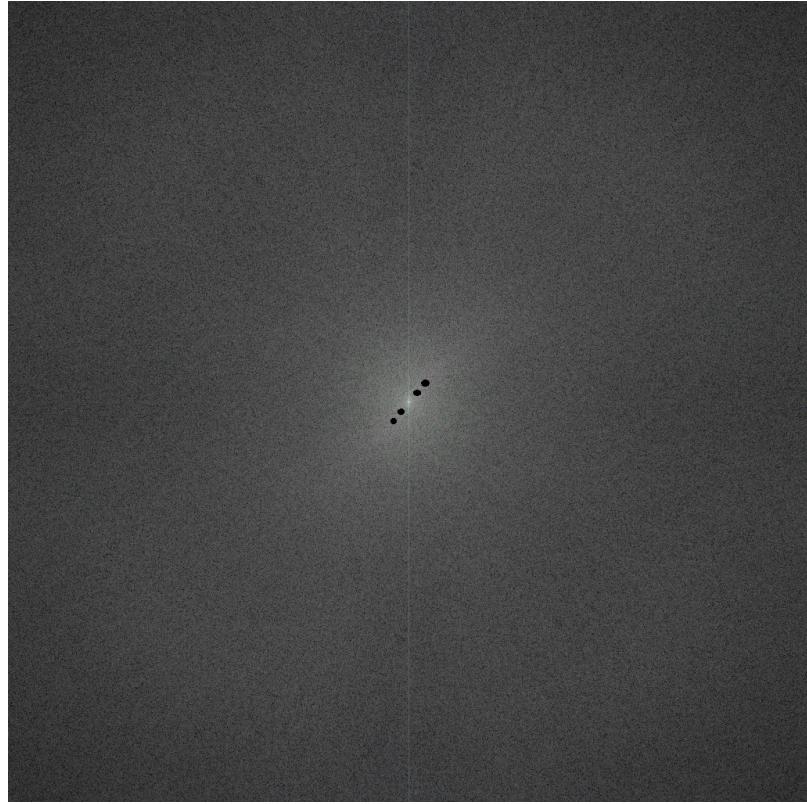
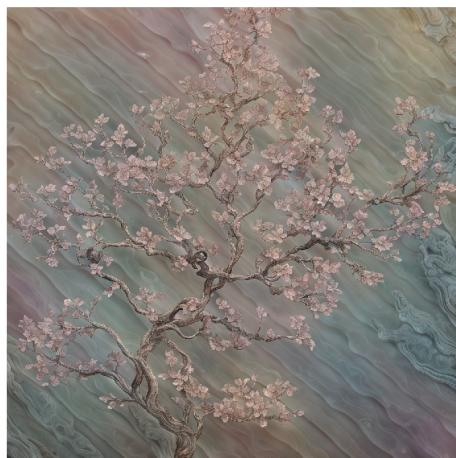


Рис. 4: Фильтрация шума на нормализованном логарифме модуля спектра



(a) Отфильтрованное изображение



(b) Оригинальное изображение

Рис. 5: Сравнение восстановленного и исходного изображений

фильтровать нежелательные пики, чтобы восстановленная картинка стала визуально чище. Однако стоит отметить, что хоть ручной метод и показал себя с эффективной стороны, он всё ещё требует очень большой аккуратности в выборе подавляемой области (то, что мы закрашиваем чёрным цветом), ведь одно малейшее движение в сторону может испортить детали изображения. Собственно, именно поэтому в реальности применяют различные маски, которые заглушают лишнее автоматически.

2 Исследование свёртки

Что ж, пришло время исследовать эффекты, которые мы можем накладывать на изображения, проделывая их свёртку с различные ядрами. Для начала найдем на просторах интернета изображение малого размера, с которым и будем в дальнейшем работать весь этот пункт, и преобразуем его в черно-белые оттенки для упрощения анализа командой **im2gray**. Проделанное представлено ниже

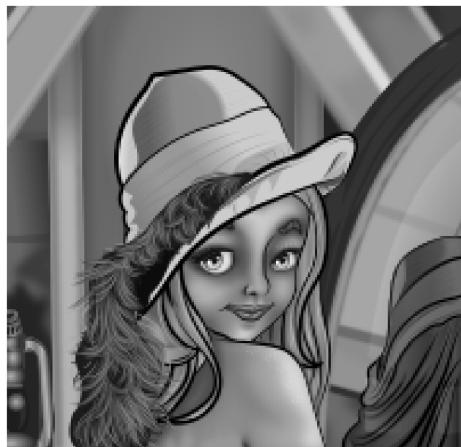


Рис. 6: Исходное изображение (оттенки серого)

2.1 Размытие по Гауссу

Мы будем изучать каждый из эффектов отдельно. Их применение возможно через использование свёртки ядра - предварительно

заданной квадратной матрицы - с изображением. Именно правильное задание матрицы, как мы увидим после, позволяет достигать необходимых эффектов.

Итак, начнём с эффекта размытия. Сформируем ядро Гауссова фильтра при $N = 7$ (нечётное значение необходимо, для того чтобы существовал центральный элемент матрицы):

$$\sigma = \frac{N - 1}{6}, \quad A_{ij} = \exp\left(-\frac{-(i - \frac{N+1}{2})^2 + (j - \frac{N+1}{2})^2}{2\sigma^2}\right), \quad i, j \in \{1 \dots N\},$$

$$K_\sigma = \frac{A}{\sum_{ij} A_{ij}}, \quad A - \text{вся матрица.}$$

Отметим, что нормировка была необходима для сохранения средней яркости всего изображения, не было пересветов, белых пятен и прочих побочных эффектов.

Используем функция **conv2** для выполнения двумерной свертки. Результаты изображены на рисунках ниже



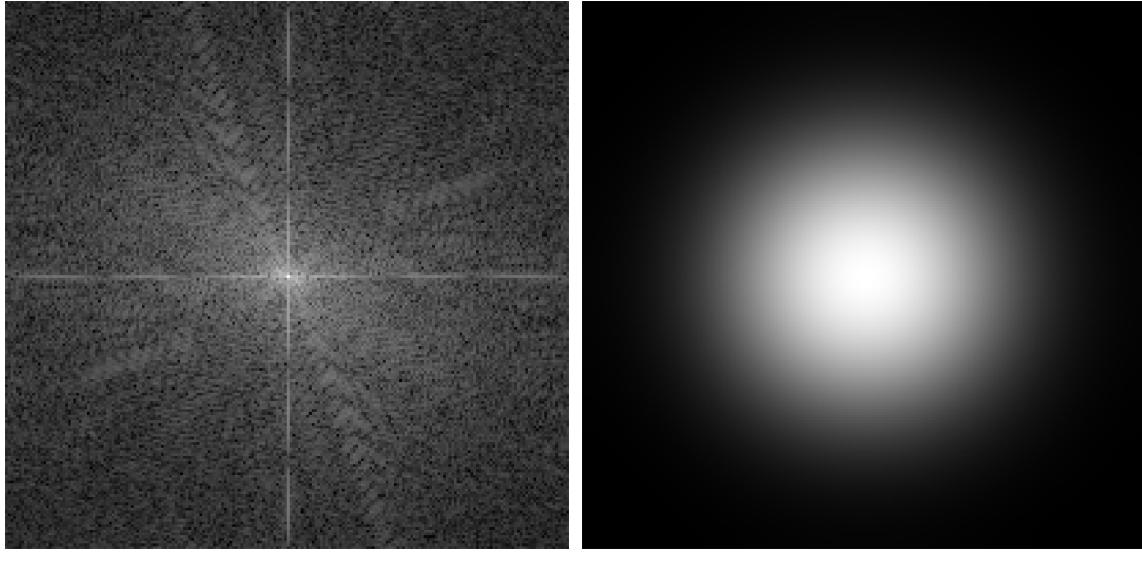
(a) Исходное изображение

(b) Размытое изображение

Рис. 7: Сравнение исходного и размытого по Гауссу ($N = 7$) изображений

Как можно видеть, полученное изображение действительно достаточно размыто, а границы смягчены, то есть свёртка с ядром работает и даёт ожидаемую картину!

Попробуем теперь иной подход: мы знаем, что свёртка во временной области эквивалентна умножению в частотной. Первое мы уже делали, испытаем второе, для этого найдём образ Фурье исходного изображения (рисунок 8а) и выбранного нами ядра (рисунок 8б).



(a) Логарифм модуля образа изображения

(b) Логарифм модуля образа ядра

Рис. 8: Модули образов изображения и ядра размытия по Гауссу $N = 7$

Спектр ядра размытия по Гауссу был расширен до размеров исследуемого изображения, и мы можем совершить поэлементное перемножение образов, получив спектр фильтрованного сигнала (рисунок 9).

Полученный результат показывает, что размытие по Гауссу подавляет высокие частоты образа изображения, действуя как низкочастотный фильтр (именно поэтому он и имеет такую округлую форму с ярким белым пятном в центре).

Теперь восстановим изображение и сравним его с полученным в предыдущем методе (рисунок 10, где оба способа выведены рядом). По итогу получаются практически идентичные результаты, что подтверждает эквивалентность операций свёртки и перемножения в частотной области (однако последнее может быть чуть более предпочтительным на больших изображениях, так как совершается меньшее число операций).

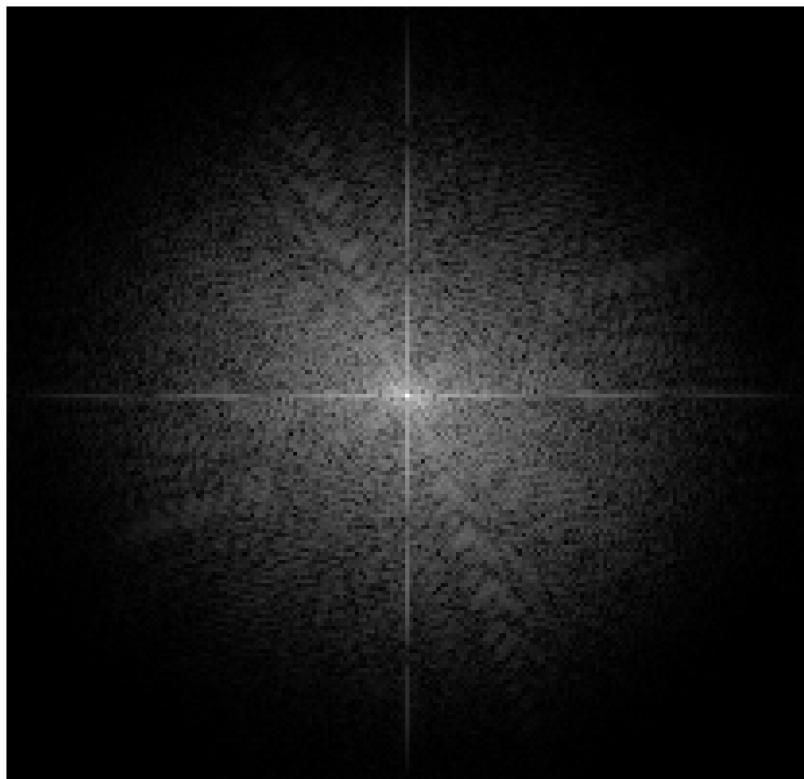
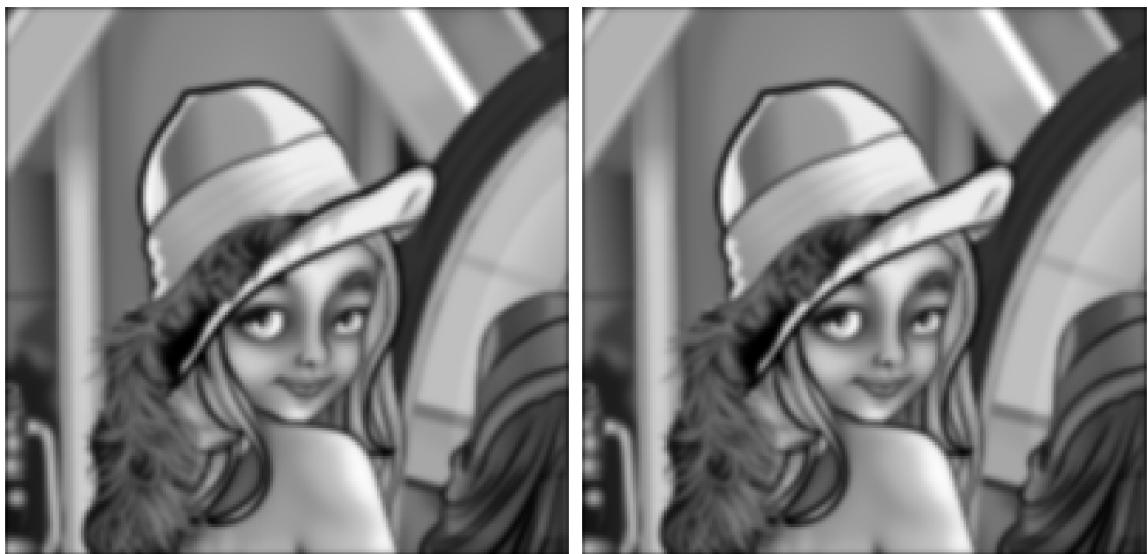


Рис. 9: Фильтрация изображения с помощью размытия Гаусса $N = 7$



(a) Размытие по Гауссу (свёртка)

(b) Размытие по Гауссу (умножение)

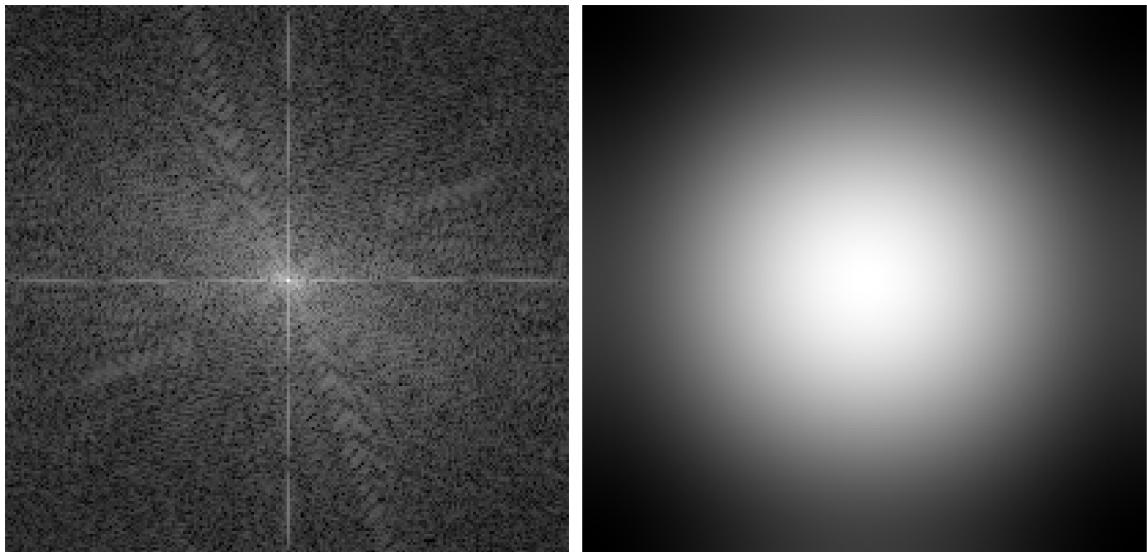
Рис. 10: Свёртка и частотное умножение при размытии по Гауссу $N = 7$



(a) Исходное изображение

(b) Размытое изображение

Рис. 11: Сравнение исходного и размытого по Гауссу ($N = 5$) изображений



(a) Логарифм модуля образа изображения

(b) Логарифм модуля образа ядра

Рис. 12: Модули образов изображения и ядра размытия по Гауссу $N = 5$

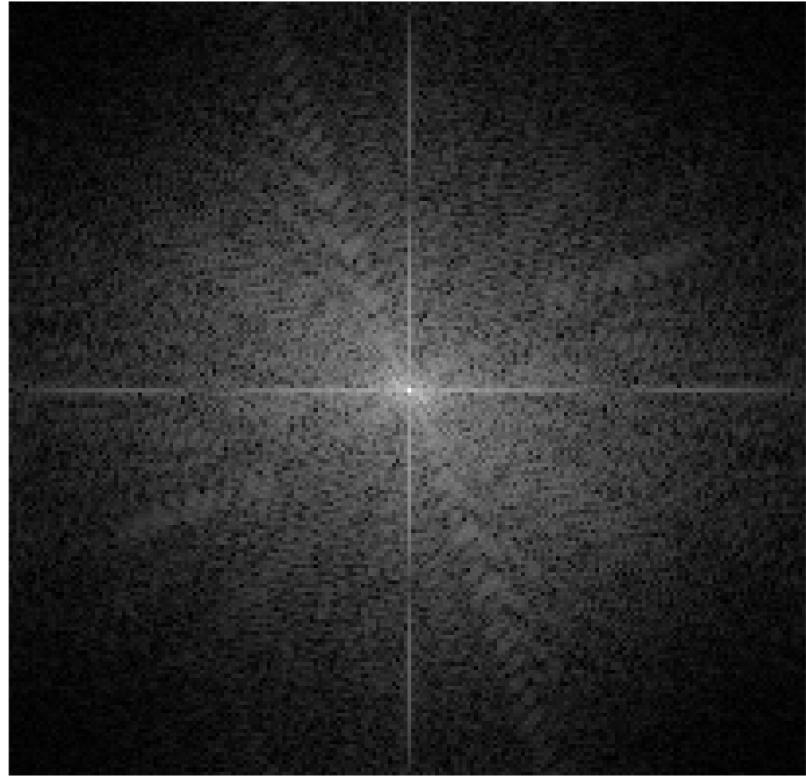


Рис. 13: Фильтрация изображения с помощью размытия Гаусса $N = 5$



(a) Размытие по Гауссу (свёртка)

(b) Размытие по Гауссу (умножение)

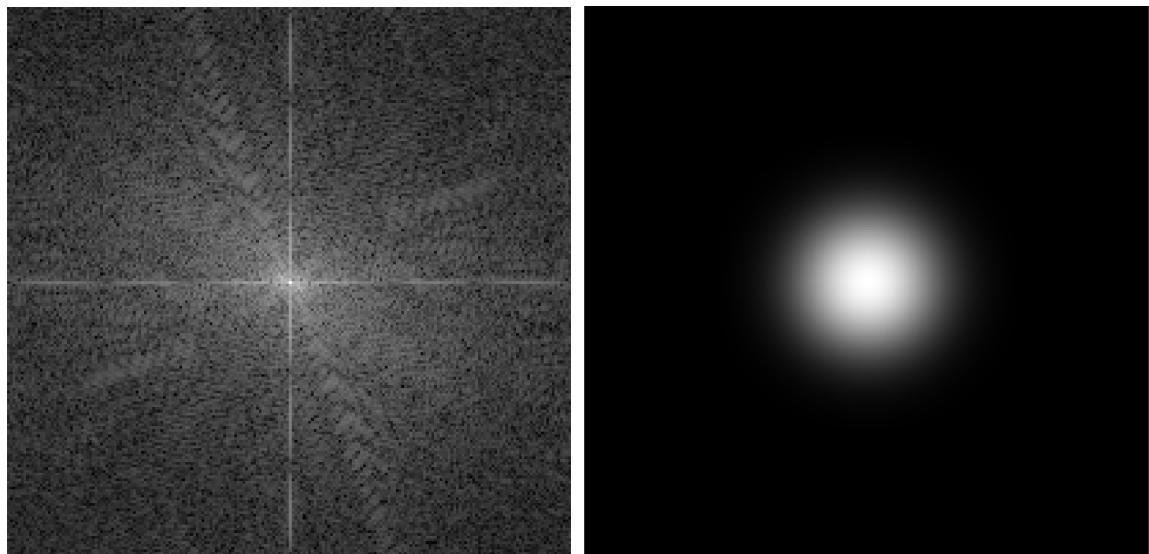
Рис. 14: Свёртка и частотное умножение при размытии по Гауссу $N = 5$



(a) Исходное изображение

(b) Размытое изображение

Рис. 15: Сравнение исходного и размытого по Гауссу ($N = 13$) изображений



(a) Логарифм модуля образа изображения

(b) Логарифм модуля образа ядра

Рис. 16: Модули образов изображения и ядра размытия по Гауссу $N = 13$

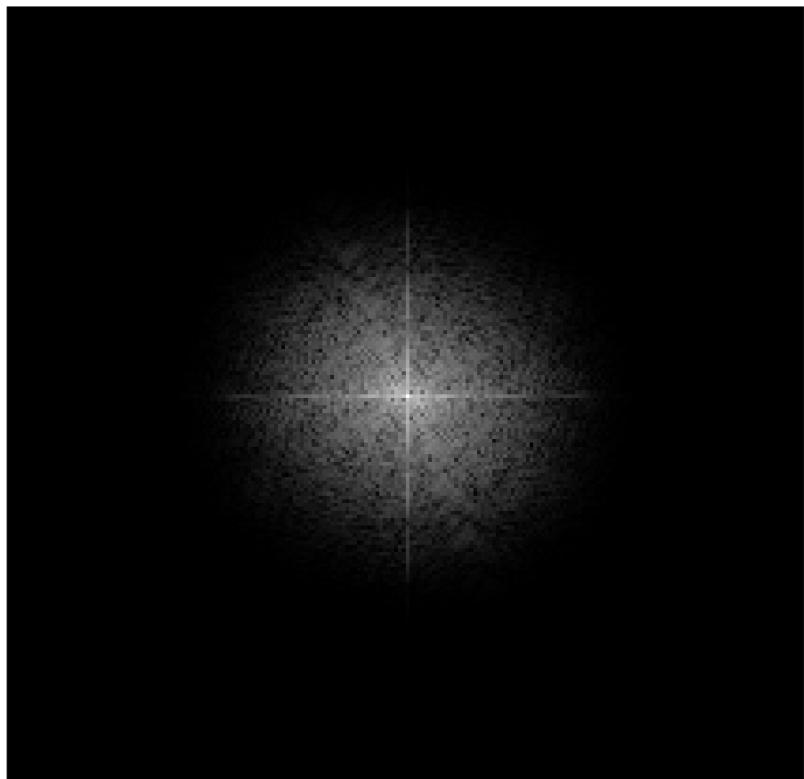
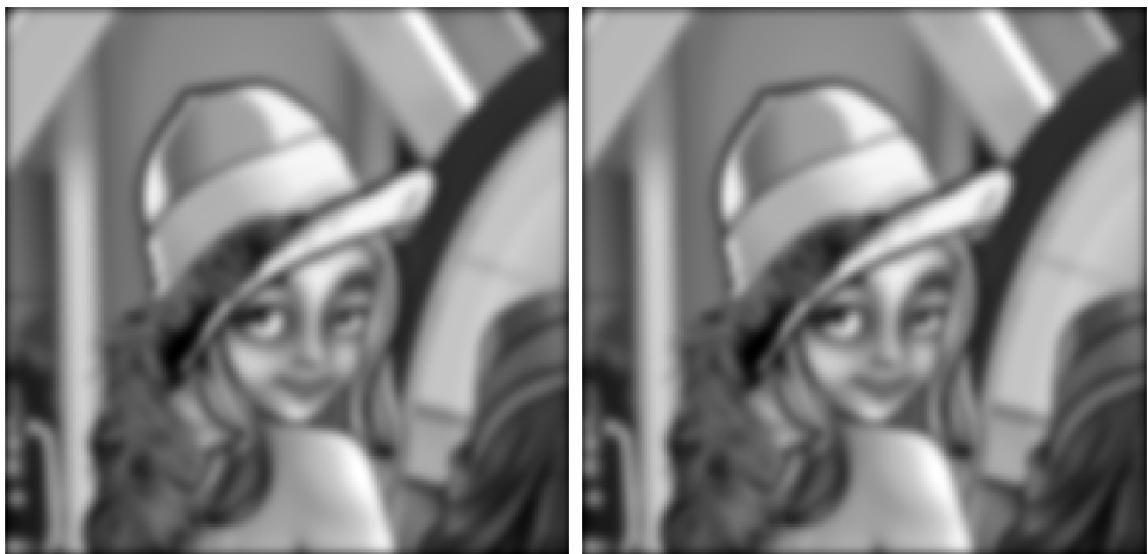


Рис. 17: Фильтрация изображения с помощью размытия Гаусса $N = 13$



(a) Размытие по Гауссу (свёртка)

(b) Размытие по Гауссу (умножение)

Рис. 18: Свёртка и частотное умножение при размытии по Гауссу $N = 13$



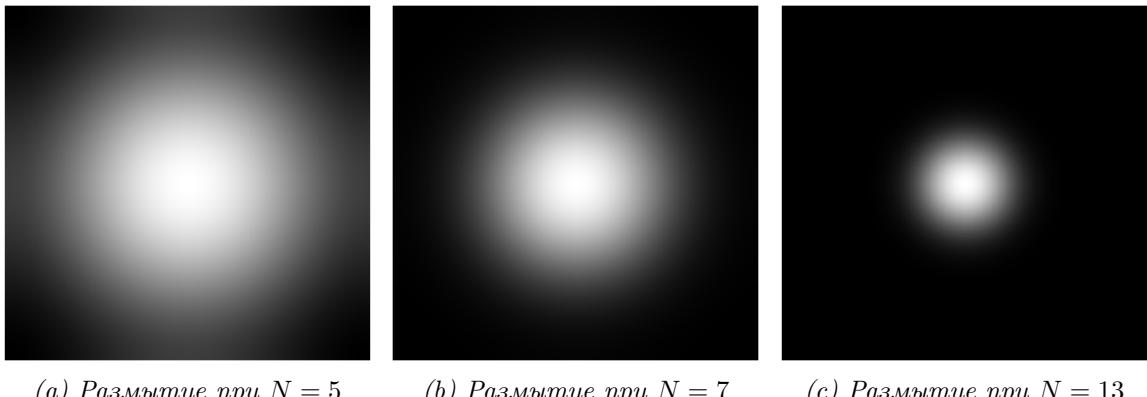
(a) Размытие при $N = 5$ (b) Размытие при $N = 7$ (c) Размытие при $N = 13$

Рис. 19: Сравнение применения размытия по Гауссу для различных N

Также были проделаны аналогичные шаги для параметров $N = 5$ (рисунки 11-14) и $N = 13$ (рисунки 15-18). Полученное говорит о том, что увеличение N влечет увеличение и $\sigma = \frac{N-1}{6}$, участвующей в формировании ядра размытия по Гауссу и задающую «силу» данного размытия. Соответственно, чем большие параметры мы задаём, тем больший эффект получаем за счёт уменьшения круга частот (рисунок 20), чьих амплитуд мы мало касаемся и оставляем практически нетронутыми (широкие области заметны при малых значениях N , в этом случае мы оставляем частотную область практически нетронутой). Получается, матрицы полностью формируют то, как будет действовать наш фильтр, и именно поэтому им уделяется такое особое внимание. По вышеприведенным результатам можно понять и то, что высокие частоты в изображении отвечают за мелкие детали, а низкие, объёмные, - за его главные составляющие (его «суть»).

2.2 Блоchное размытие

В качестве альтернативного фильтра применим блочное размытие. Для начала зададим ядро с помощью матрицы размера 5×5 ($N = 5$), заполненной единицами, и отнормируем её. Это означает, что свёртка будет вычислять простое среднее значение по каждому



(a) Размытие при $N = 5$ (b) Размытие при $N = 7$ (c) Размытие при $N = 13$

Рис. 20: Сравнение логарифмов модулей образов Фурье размытий по Гауссу для различных N

квадрату пикселей. Математически результат можно записать так:

$$A_{ij} = 1, \quad i, j \in \{1, \dots, N\}, \quad K_{\square} = \frac{A}{\sum_{i,j} A_{ij}}, \quad N = 5.$$

Осуществим операцию свёртки аналогичной предыдущему пункту функцией **conv2** и получим изображение на рисунке 21b.



(a) Исходное изображение

(b) Размытое изображение

Рис. 21: Сравнение исходного и блочно размытого ($N = 5$) изображений

Так же, как и прежде, проверим теорему о свёртке. Для перехода к частотной области воспользуемся двумерным быстрым преобразованием Фурье, а для лучшего восприятия воспользуемся логарифмической шкалой, позволяющей усилить контраст между различными частотными компонентами. Найденные образы показаны на рисунках 22 и 23 (для фильтрации было применено поэлементное умножение спектров изображения и ядра).

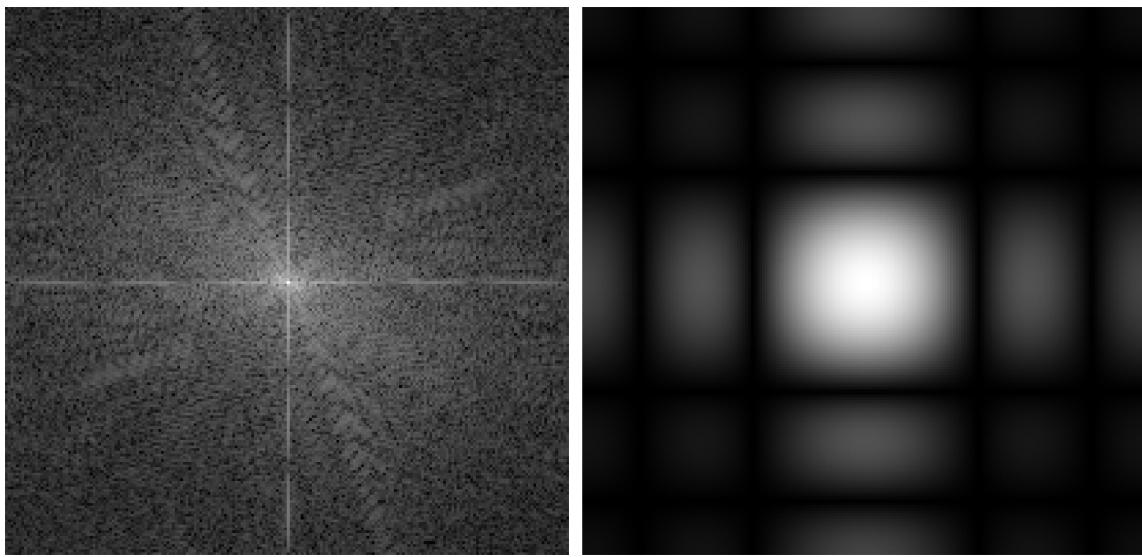


Рис. 22: Модули образов изображения и ядра блочного размытия при $N = 5$

Видим, что образ ядра представляет из себя фильтр, всё так же глушащий высокие частоты, только чуть иным в сравнении с Гауссовым размытием способом: в первом случае идёт пропуск в заданном квадратном диапазоне (к тому же имеются некоторые периодические прямоугольные «выделения» на высоких частотах, а это означает, что глушатся они всё же не полностью), во втором - в круге.

Восстановление изображения из его отфильтрованного Фурье-образа даёт аналогичные свёртке результаты (рисунок 24, сравнивающий эти два метода подхода к получению эффектов на изображениях), что в очередной раз подтверждает изученную нами теорему о свёртке, показывающую эквивалентность операций этой самой свёртки во временной области и умножения в частотной.

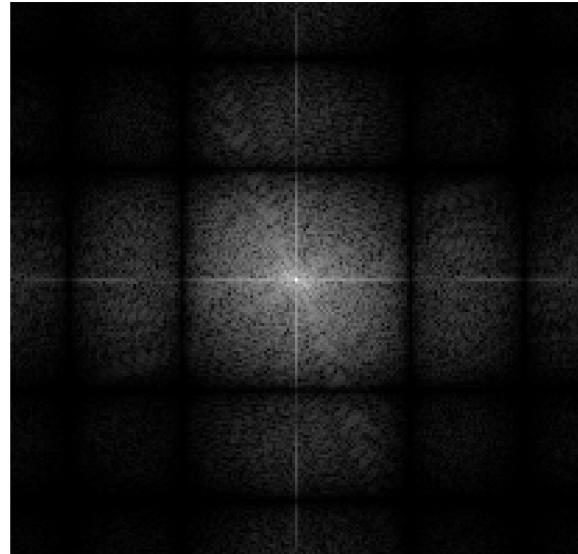


Рис. 23: Фильтрация изображения с блочным размытием при $N = 5$



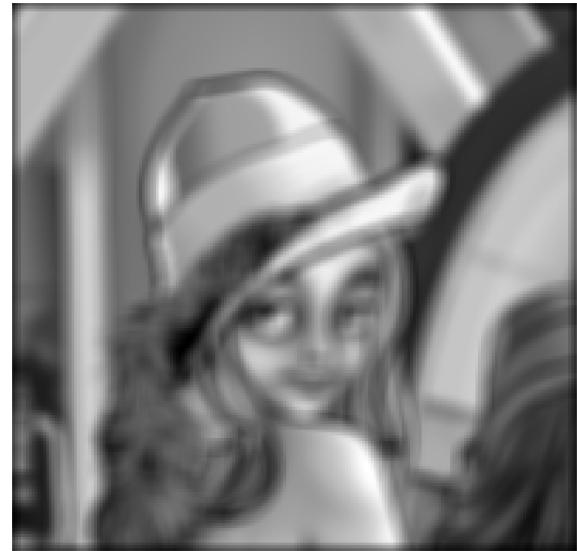
(a) Блочное размытие (свёртка)

(b) Блочное размытие (умножение)

Рис. 24: Свёртка и частотное умножение при блочном размытии при $N = 5$

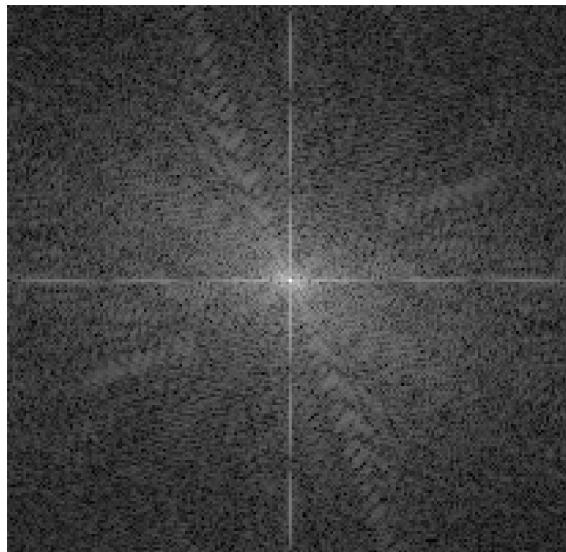


(a) Исходное изображение

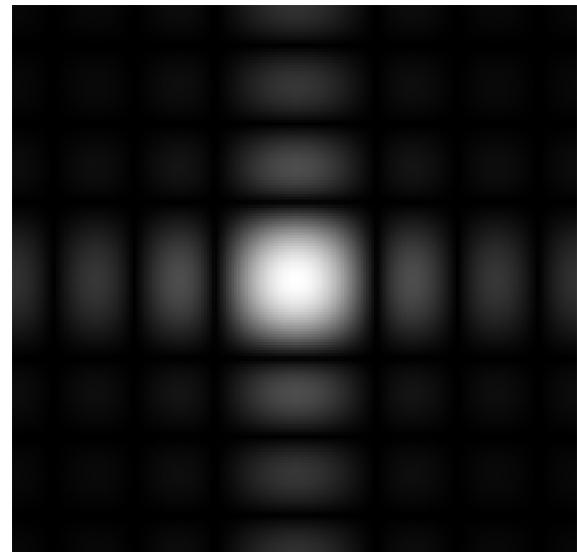


(b) Размытое изображение

Рис. 25: Сравнение исходного и блочно размытого ($N = 7$) изображений



(a) Логарифм модуля образа изображения



(b) Логарифм модуля образа ядра

Рис. 26: Модули образов изображения и ядра блочного размытия при $N = 7$

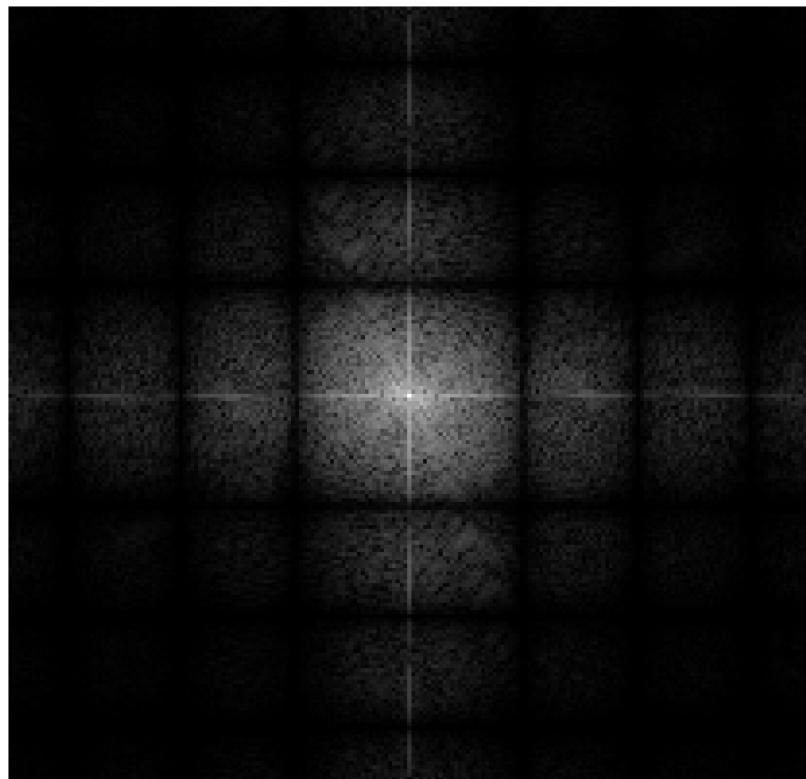
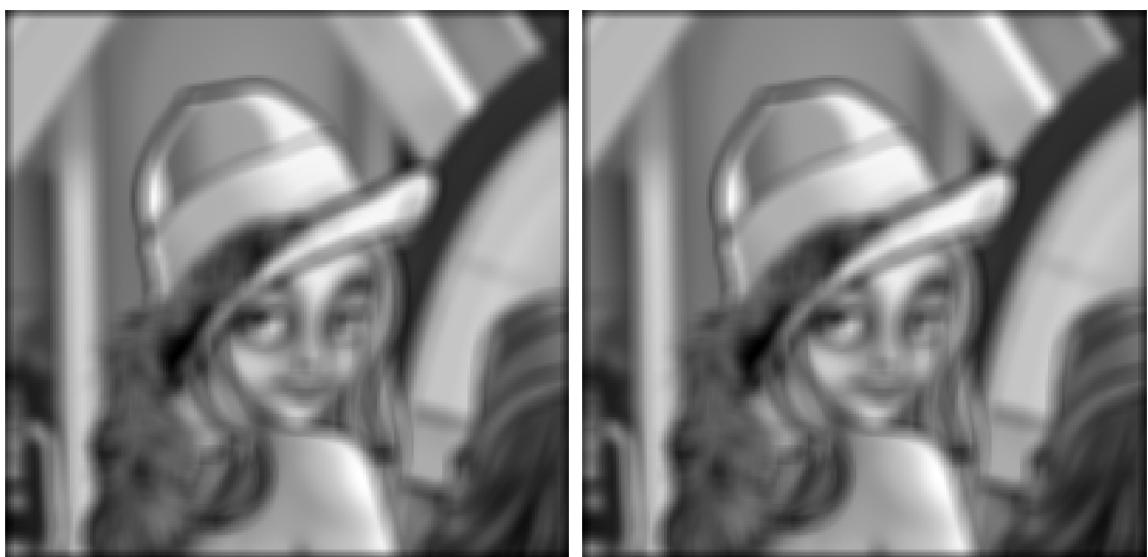


Рис. 27: Фильтрация изображения с блочного размытия при $N = 7$



(a) Блочное размытие (свёртка)

(b) Блочное размытие (умножение)

Рис. 28: Свёртка и частотное умножение при блочном размытии при $N = 7$

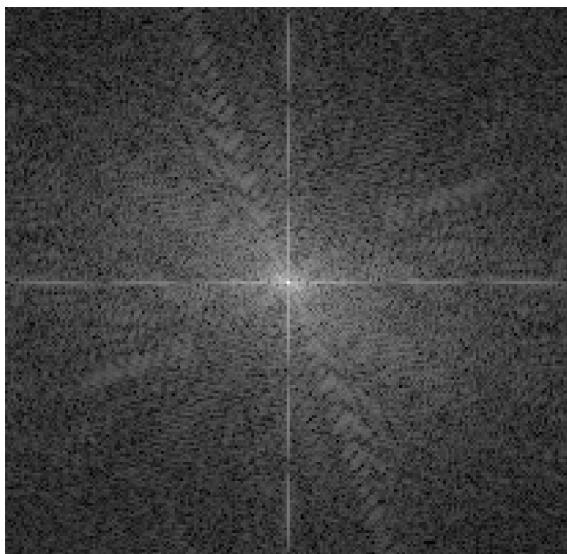


(a) Исходное изображение

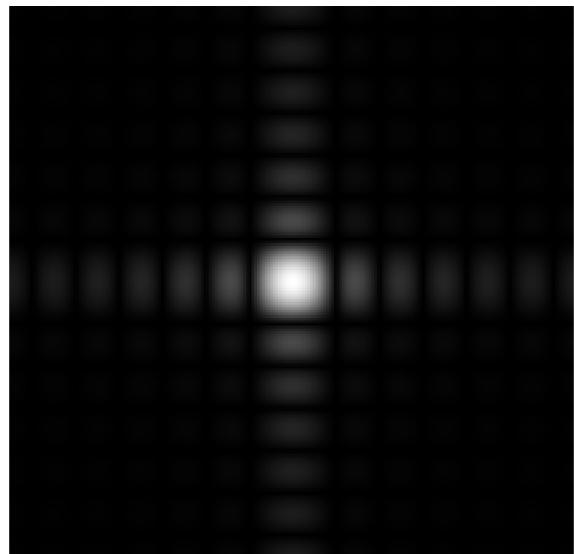


(b) Размытое изображение

Рис. 29: Сравнение исходного и блочно размытого ($N = 13$) изображений



(a) Логарифм модуля образа изображения



(b) Логарифм модуля образа ядра

Рис. 30: Модули образов изображения и ядра блочного размытия при $N = 13$

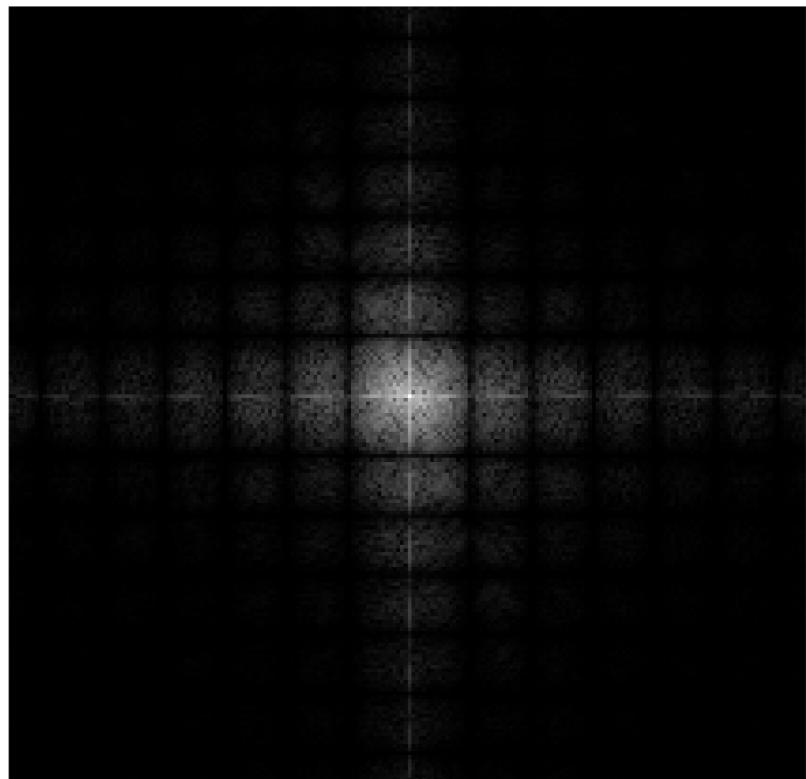
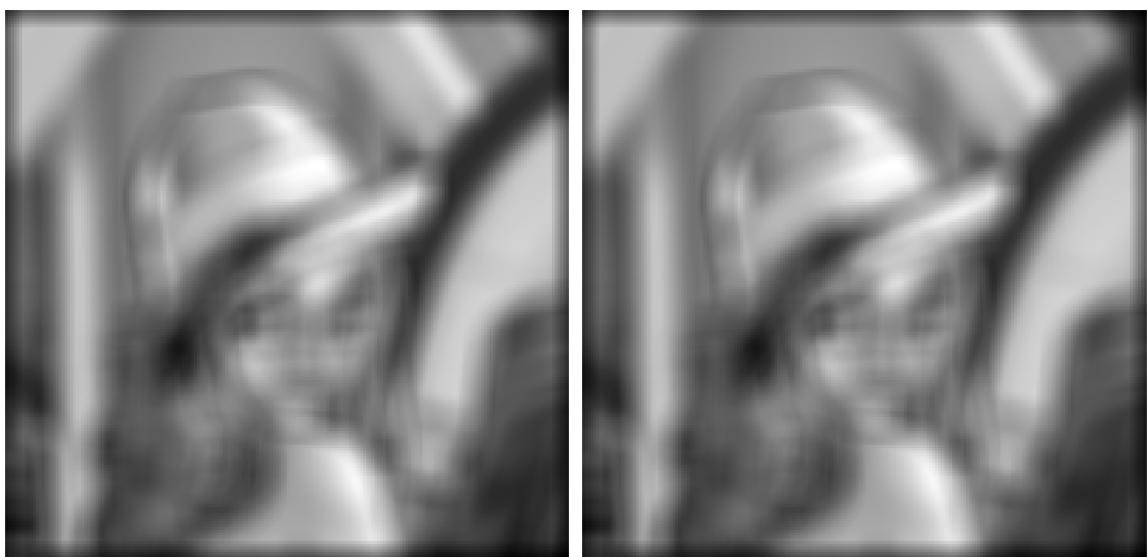


Рис. 31: Фильтрация изображения с блочным размытием при $N = 13$



(a) Блочное размытие (свёртка)

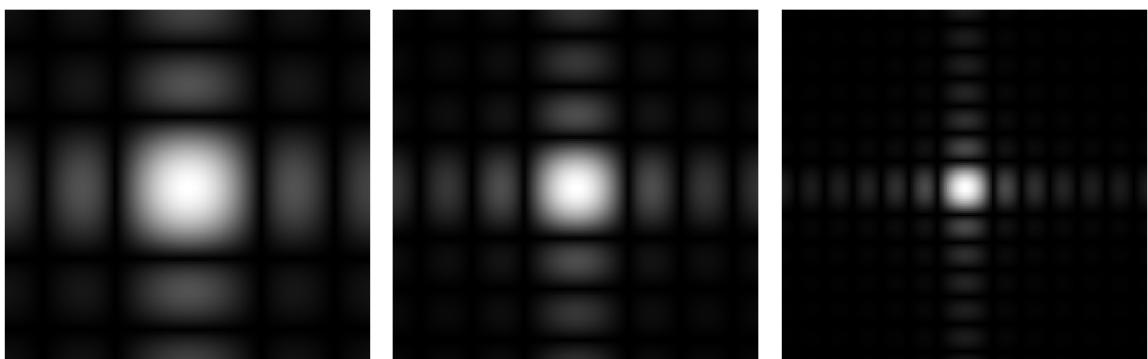
(b) Блочное размытие (умножение)

Рис. 32: Свёртка и частотное умножение при блочном размытии при $N = 13$



(a) Размытие при $N = 5$ (b) Размытие при $N = 7$ (c) Размытие при $N = 13$

Рис. 33: Сравнение применения блочного размытия для различных N



(a) Размытие при $N = 5$ (b) Размытие при $N = 7$ (c) Размытие при $N = 13$

Рис. 34: Сравнение логарифмов модулей образов блочного размытий для N

(a) Размытие по Гауссу $N = 13$ (b) Блоchное размытие $N = 7$

Рис. 35: Сравнение фильтров размытия

Шаги были также проделаны для значений $N = 7$ (рисунки 25 - 28) и $N = 13$ (рисунки 29 и 32). Увеличение размера матрицы также приводит к снижению пропускаемой области, более сильно давятся верхние частоты, а значит, сильней происходит фильтрация. Также были проверены и соотнесены результаты наложения эффекта через свёртку и умножения образов, всё оказалось равным друг другу.

На рисунке 35 сравниваются исследованные нами фильтры размытия. Заметно, что блочное размытие при тех же N является более сильным в сравнении с Гауссовым (так как даже при $N = 13$ размытие по Гауссу находится на уровне блочного при $N = 7$), но приводит к более резкому виду границ и менее естественному результату (в том числе и из-за образовывающихся прямоугольных «окон» на высоких частотах образа Фурье ядра). Всё дело в том, что в отличие от блочного, размытие по Гауссу придаёт пикселям, расположенным ближе к центру, больший вес. Это создаёт гладкий переход и помогает в естественности, и именно поэтому Гауссовский фильтр предпочтителен в задачах предварительной обработки изображений (однако блочное размытие всё ещё может пригодится, например, если необходимо достичь большего эффекта за меньшее число операций).

2.3 Увеличение резкости

Отлично, с размытием разобрались. Переайдём теперь к другим доступным возможностям операции свёртки. На очереди - ядро повышения резкости, задающееся матрицей

$$K_* = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

Как и прежде, посмотрим, на что введённое способно, как раньше, загрузим всё то же изображение в matlab и с помощью функции **conv2** найдём его свертку с ядром K_* . Получим следующее:



(a) Исходное изображение

(b) Изображение с увеличенной резкостью

Рис. 36: Сравнение исходного изображения и с увеличенной резкостью

На отфильтрованном изображении улучшилось чёткость контуров (шляпы, волосы, перо стали более выраженным), также повысилась контрастность вблизи границ. Однако проявился и некий побочный недостаток в виде общей серости изображения.

Проверим теперь теорему о свёртке, для этого проделаем уже знакомые нам шаги по нахождению образов, их модулей, а также

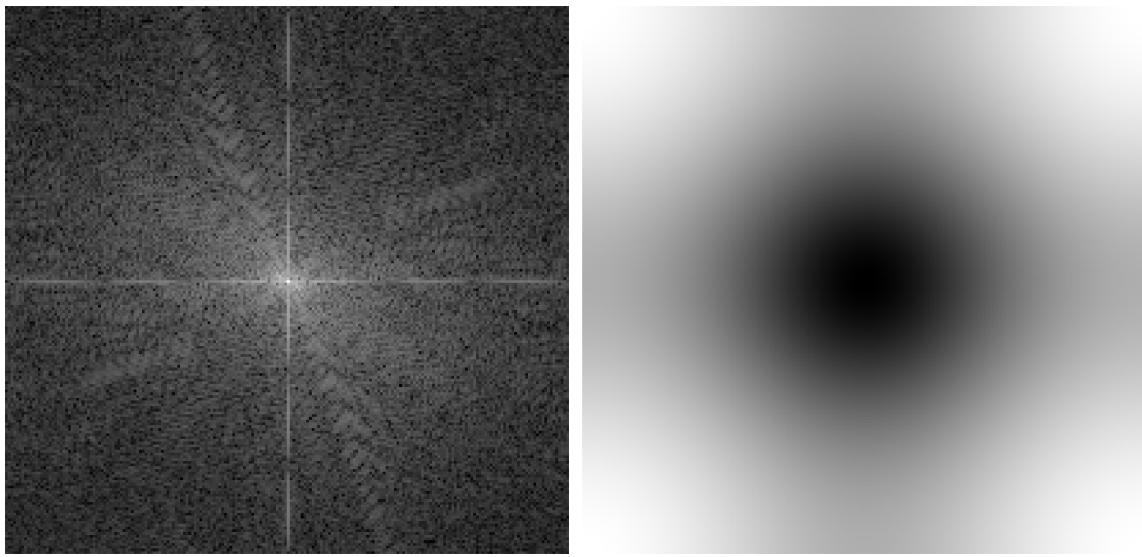


Рис. 37: Модули образов изображения и ядра повышения резкости

произведения спектров ядра и изображения. Все результаты изображены на рисунках 37, 38 и 39. Можно видеть, что всё предполагаемое в действительности выполняется.

Вид модуля образа ядра повышения резкости представляет из себя тёмный круг со светлым фоном. То есть картина противоположна сглаживанию - мы подавляем низкие частоты, находящиеся около центра, и улучшаем действие высоких частот, отвечающих, как уже было отмечено, за мелкие детали изображения (в том числе контуры). Образ Фурье ядра также объясняет и возникший побочное явление «серости» в фильтрованном изображении - так как все главные цветовые составляющие находятся возле точки $(0, 0)$ частотной области, а мы эту область подавляем, то немного искажается и цветовая гамма исходного изображения.

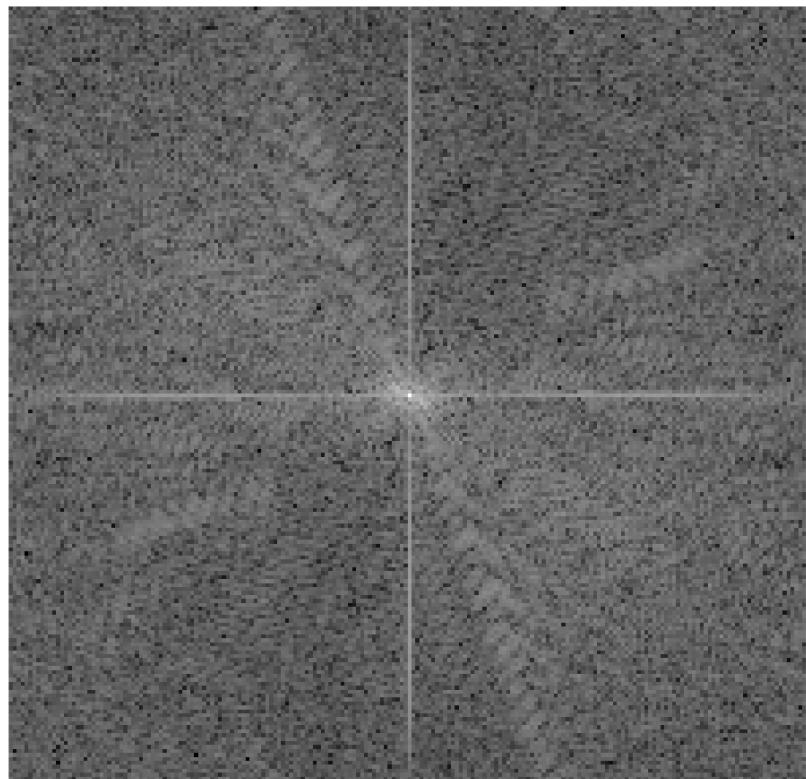


Рис. 38: Фильтрация изображения (повышение резкости)



(a) Фильтрация через свёртку

(b) Фильтрация через умножение

Рис. 39: Свёртка и частотное умножение при ядре, повышающем резкость

2.4 Выделение краёв

Испытаем наш аппарат в контексте применения эффекта выделения краёв, для зададим ядро свёртки через матрицу

$$K_{\nabla} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

Введенное ядро усиливает центральный пиксель относительно его соседей, тем самым выделяя, как мы увидим далее, резкие изменения яркости и границы объектов. Применим свёртку ядра, используя обычный метод:



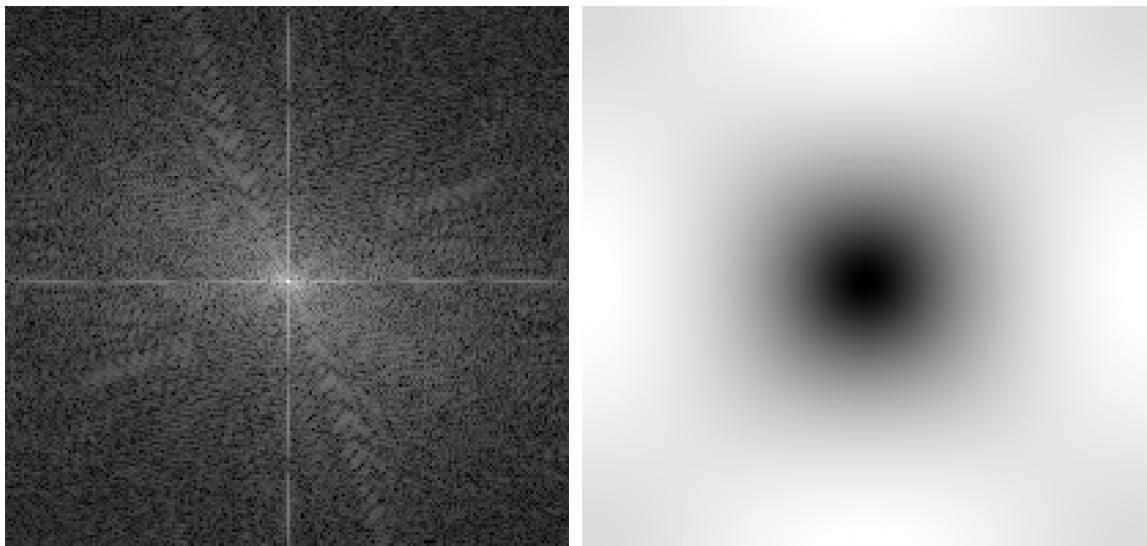
(a) Исходное изображение

(b) Изображение с выделенными краями

Рис. 40: Сравнение исходного изображения и с выделенными краями

Итак, полученное изображение обладает яркими, определёнными границами и контрастными переходами. Шляпа, силуэт девушки и его отражение в зеркале, волосы - всё обладает отчётливым и ярко выраженным контуром. Так что фильтрованное изображение в действительности задаёт выделение краёв.

Посмотрим на непрямой метод получения эффектов - через умножение в частотной области образов Фурье изображения и ядра:



(a) Логарифм модуля образа изображения

(b) Логарифм модуля образа ядра

Рис. 41: Модули образов изображения и ядра выделения краёв

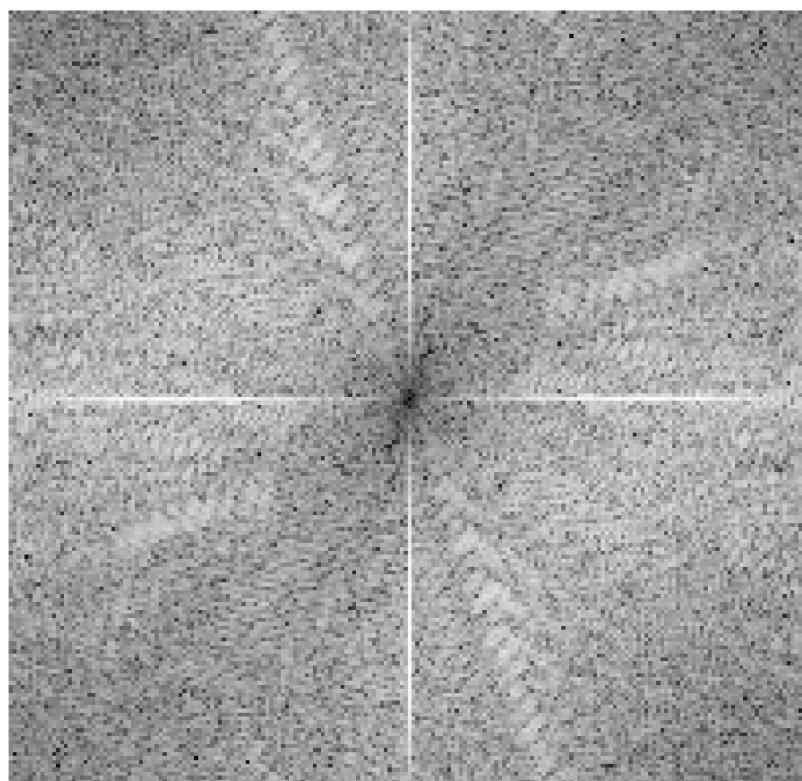


Рис. 42: Фильтрация изображения (выделение краёв)



(a) Фильтрация через свёртку

(b) Фильтрация через умножение

Рис. 43: Свёртка и частотное умножение при ядре, выделяющего края

На рисунке 43 сравниваются пространственный и частотный методы применения эффекта к изображению. Как видим, они дают практически одинаковые результаты, а значит выполняется теорема о свёртке. По рисунку 41b же можно понять, что ядро выделения краёв работает как фильтр низких частот (равно как и увеличения резкости, только тут достигается меньший объём и большая концентрация тёмного пятна в центре, снижающего влияние низких частот, проявляющихся в общих паттернах (макро-детали) на изображении; белый фон же наоборот работает на высокие компоненты, влияющие такие микро-детали, как контуры, поэтому-то рассматриваемое ядро и работает соответствующим образом).

2.5 Своё ядро

До этого мы изучали уже данные нам ядра. Как можно было понять, они полностью задаются матрицами, с которыми после мы проделываем свёртки, находим модули образов, проводим фильтрации в частотной области, дающие эквивалентные результаты и немногое другое. Именно матрицы влияют на то, каким будет наш эффект.

Так давайте создадим свой, опять-таки через ядра:

$$G = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{bmatrix},$$

который мы не медля испытаем в действии:



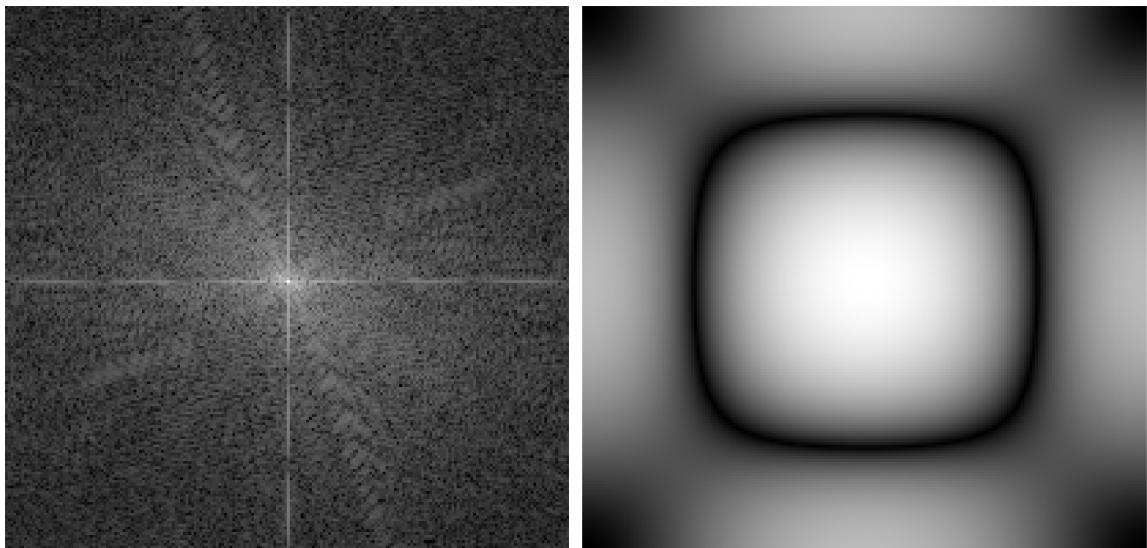
(a) Исходное изображение

(b) Изображение после свёртки

Рис. 44: Сравнение исходного изображения и свёрнутого с собственным ядром

На рисунке 44 заметны эффекты, которые даёт свёртка с ядром, - самое заметное то, что однотонные части изображения (шляпа, перо, цвет лица) стали тёмными, а все контуры были инвертированы: тёмные тона обводок стали светлыми, вышла «негативная маска краёв».

Погрузимся детальнее в суть и посмотрим, что же происходит в частотной области (рисунки 45-47); теорема о свёртке работает. Область логарифма модуля образа ядра возле «0» светлая, ярко белая, а значит, главные паттерны изображения остаются неизменными. В то же время с какой-то (малой) степенью давятся высокочастотные компоненты («в краях»), а где-то посередине образуется область тёмного оттенка, то есть хорошо тушится определённый отрезок средних частот. Всё это вкупе и даёт то отфильтрованное изображение с ин-



(a) Логарифм модуля образа изображения

(b) Логарифм модуля образа ядра

Рис. 45: Модули образов изображения и собственного ядра

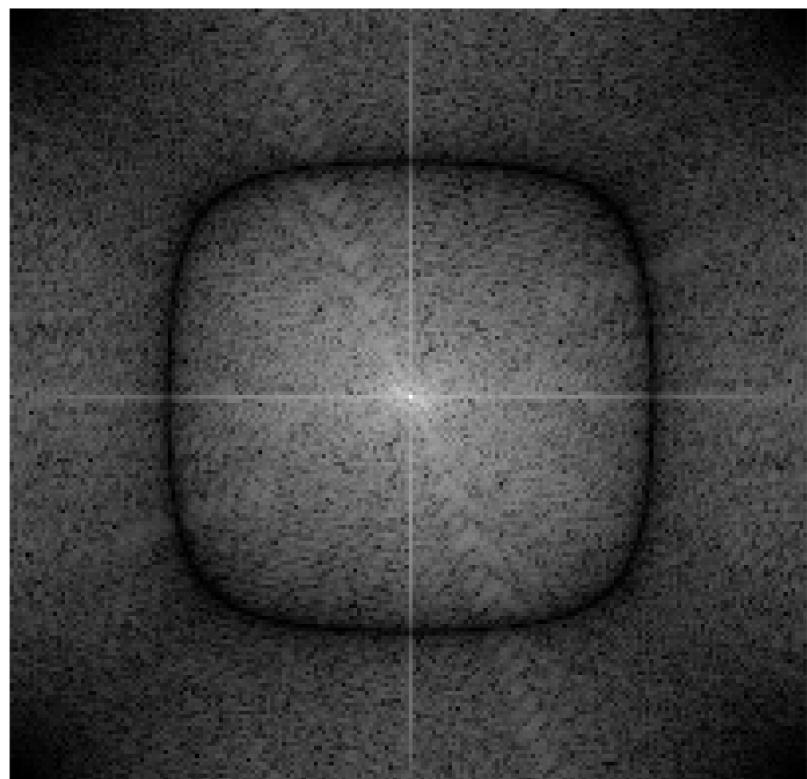


Рис. 46: Фильтрация изображения (собственное ядро)



(a) Фильтрация через свёртку

(b) Фильтрация через умножение

Рис. 47: Свёртка и частотное умножение при собственном ядре

версиями (которые существуют за счёт нулевого вклада рассматриваемого пикселя и обратных значений расположенных остальных), которое мы и получаем на выходе.

3 Общие выводы

В ходе лабораторной работы было выяснено, что фильтрация изображений в частотной области позволяет эффективно устранять периодические шумы, проявляющиеся в виде дельта-пиков на спектре. Применение свёртки с различными ядрами показало разнообразные эффекты: гауссово и блочное размытие сглаживают изображение, но гауссов фильтр обеспечило более естественный результат. Ядра повышения резкости, выделения краёв и контрастности позволили усилить детали и структуру изображения, подчеркнуть границы объектов. Сравнение прямого и частотного подходов к свёртке подтвердило их эквивалентность, однако частотный метод с перемножением образов Фурье может быть вычислительно более выгоден для больших изображений за счёт меньшего числа совершаемых операций.

4 Приложение

Листинг 1: Сохранение логарифмированного модуля Фурье-образа изображения

```
1 temp_initial = imread("toedit.png");
2 imshow(temp_initial, [])
3 img_initial = double(temp_initial)/255;
4 fourier = fftshift(fft2(img_initial));
5 ab = abs(fourier);
6 an = angle(fourier);
7 l = log(ab+1);
8 m = max(l(:));
9 ul = l/m;
10 imwrite(ul, "log.png");
```

Листинг 2: Код для восстановления отфильтрованного изображения

```
1 temp_out = imread("log_edited.png");
2 img_out = double(temp_out)/255;
3 abs_out = exp(img_out*m) - 1;
4 fourier_out = abs_out.*exp(1i.*an);
5 img_prepared = ifft2(ifftshift(fourier_out));
6 %imwrite(img_prepared, "edited.png");
7 imshow(img_prepared, []);-
```

Листинг 3: Код для исследования ядра, дающего размытие по Гауссу

```

1 img = imread('1.png');
2 gray_img = im2gray(img);
3 gray_img = im2double(gray_img);
4
5 N = 13;
6 sigma = (N - 1) / 6;
7 center = (N + 1) / 2;
8
9 [X, Y] = meshgrid(1:N, 1:N);
10 G = exp(-((X - center).^2 + (Y - center).^2) / (2 * sigma^2));
11 G = G / sum(G(:));
12
13 conv_img = conv2(gray_img, G, 'same');
14
15 [h, w] = size(gray_img);
16 [hk, wk] = size(G);
17 H = fft2(gray_img, h + hk - 1, w + wk - 1);
18 Gf = fft2(G, h + hk - 1, w + wk - 1);
19
20 filtered = H .* Gf;
21 result_fft = real(ifft2(filtered));
22 start_row = floor((hk - 1) / 2) + 1;
23 start_col = floor((wk - 1) / 2) + 1;
24 result_fft = result_fft(start_row:start_row+h-1, ...
    start_col:start_col+w-1);
25
26 fourier = fftshift(filtered);
27 ab = abs(fourier);
28 an = angle(fourier);
29 l = log(ab+1);
30 ul = l/m;
```

Листинг 4: Код для остальных ядер (общая структура, как в предыдущем)

```

1 %N = 13;
2 %G = ones(N, N);
3 %G = G / sum(G(:));
4
5 %G = [0 -1 0; -1 5 -1; 0 -1 0];
6
7 %G = [-1 -1 -1; -1 8 -1; -1 -1 -1];
8
9 G = [-1 -1 -1; -1 0 -1; -1 -1 -1];
```