

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**УНИВЕРСИТЕТ ИТМО**

**А.С. Васильев, А.В. Васильева**

# **ОБРАБОТКА ОПТИЧЕСКИХ ИЗОБРАЖЕНИЙ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

## **Учебно-методическое пособие**

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО  
по направлению подготовки (специальности) 12.04.02 – «Оптотехника»,  
27.04.05 «Инноватика» и 12.05.01 – «Электронные и оптико-электронные  
приборы и системы специального назначения» в качестве  
учебно-методического пособия для реализации основных  
профессиональных образовательных программ высшего образования  
магистратуры и специалитета

 **УНИВЕРСИТЕТ ИТМО**

**Санкт-Петербург**

**2017**

Васильев А.С., Васильева А.В. Обработка оптических изображений. Лабораторный практикум. – СПб: Университет ИТМО, 2017. – 47 с.

Рецензенты:

Цыганок Е.А., к.т.н., доцент каф. ПиКО, Университет ИТМО

Смирнова И.Г., к.т.н., доцент каф СТО, Университет ИТМО

Пособие направлено на ознакомление и изучение основных методов обработки цифровых изображений приобретение навыков применения этих методов посредством процесса поэтапной обработки изображений. В практикуме приведены 6 лабораторных работы направленных на изучение основных методов и алгоритмов цифровой обработки изображений с использованием пакета системы Image Processing ToolBoxes, входящего в среду программирования MATLAB.

Учебное пособие предназначено для студентов по направлению подготовки бакалавров и магистров 12.04.02 – «Опготехника», 27.04.05 «Инноватика» и по специальности 12.05.01 – «Электронные и оптико-электронные приборы и системы специального назначения».



**Университет ИТМО** – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2017

©Васильев А.С., Васильева А.В.. 2017

## **Введение**

Не менее 90% всей информации человек получает при помощи зрения. Более того, объемы зрительной информации, которую приходится получать, анализировать и воспринимать, постоянно возрастают. Все большая роль при этом отводится системам технического зрения и обработки видеоданных, реализующим методы обработки изображений для последующего представления человеку, а также для автоматического анализа и выбора последующей реакции. Число областей, в которых исходные данные поступают в виде изображений, огромно. К ним относятся различные системы наблюдения, технического зрения, мониторинга, автономные системы, полиграфия, медицина, Интернет и многие другие направления.

В связи с этим значительно увеличилась роль программных средств обработки изображений. Они предоставляют богатые возможности преобразования и улучшения данных. Для понимания процесса улучшения изображения, объяснения какие именно операции производятся над изображением, определения алгоритма и последовательности необходимых действий для достижения поставленной цели. Решение этих задач возможно только на основе научного подхода, который предлагается реализовать в виде лабораторных работ.

## **1 Базовые функции для работы с цифровым изображением в среде MATLAB**

Среда MATLAB является языком высокого уровня для выполнения технических и научных вычислений. В ней встроены инструменты для выполнения вычислений, визуализации и программирования в удобной пользовательской среде. Отличительная особенность MATLAB заключается в том, что базовым элементом выступает массив элементов, который не требует задания фиксированной размерности. Это позволяет легко формулировать условия и решения многих вычислительных задач, которым требуется матричное представление объектов (что особенно ценно для обработки цифровых изображений). При этом необходимая работа займет небольшую долю времени, которое потребовалось бы для написания аналогичной программы на других языках программирования в скалярном виде. MATLAB широко используется в различных областях промышленности, техники и науки.

Система MATLAB имеет расширения в виде наборов специализированных программ, которые называются «toolbox». Одним из таких наборов является пакет Image Processing Toolbox (IPT), который расширяет возможности стандартной среды MATLAB для решения задач цифровой обработки изображений. Другие наборы, которые часто применяются для обработки изображения – это пакеты Signal Processing Toolbox (пакет обработки сигналов), Neural Network Toolbox (пакет для нейронной сети), Fuzzy Logic Toolbox (пакет нечеткой логики) и Wavelet Toolbox (пакет для работы с вейвлетами). В рамках выполнения лабораторных работ по дисциплине «Обработка оптических изображений» нам понадобятся функции, входящие в пакет IPT.

Пакет IPT поддерживает различные операции обработки изображений, включая

- пространственные преобразования изображений;
- морфологические операции;
- скользящую и блочную обработку;
- линейную фильтрацию различными фильтрами;
- анализ и улучшение изображений;
- восстановление изображений;
- удаление размытостей;
- обработка области интереса.

Многие функции пакета представляются в системе MATLAB в виде m-файлов. В них реализованы наиболее известные алгоритмы обработки

изображений. Также существует возможность просмотра программных кодов функций, которые реализует данные алгоритмы.

### 1.1 Считывание изображений

Для считывания (загрузки) изображения в рабочее пространство MATLAB используется функция `imread`:

```
f=imread('filename');
```

Функция `f=imread('filename')` считывает из файла с именем `filename` бинарное, полутоновое или полноцветное изображение и помещит его в массив `f`. Параметр `filename` является строкой. В Таблице 1 приведены основные форматы графических файлов, которые могут быть прочитаны функцией `imread` (или записаны функцией `imwrite`).

Таблица 1 – Основные форматы графических файлов, поддерживаемые MATLAB

Формат изображения	Расшифровка сокращения	Допустимые изображения	Глубина цвета
TIFF	Tag Image File Format	.tif, .tiff	1, 8, 24
JPEG	Joint Photographic Experts Group	.jpg, .jpeg	8, 24
GIF	Graphics Interchange Format	.gif	
BMP	Windows Bitmap (BMP)	.bmp	1, 4, 8, 24
PNG	Portable Network Graphics	.png	1, 8, 24

Для чтения информации о графическом файле используется функция `imfinfo`:

```
info_f = imfinfo('filename');
```

В структуре `info_f` возвращается информация об изображении и способе его хранения из файла с именем `filename`. Структуры для разных форматов файлов отличаются друг от друга. Общие для всех форматов первые 9 полей структуры, по которым можно определить формат файла, тип и размеры изображения, приведены в Таблице 2.

В файлах форматов TIFF и HDF может храниться несколько изображений. В этом случае `info_f` является массивом структур.

Таблица 2 – Основные параметры файла-изображения в структуре функции `imfinfo`

Имя поля	Тип данных	Описание
Filename	Строка	Имя файла, если файл находится в текущей директории, или полный путь к файлу
FileModDate	Строка	Дата и время последней модификации файла
FileSize	Число	Размер файла в байтах
Format	Строка	Формат файла
FormatVersion	Строка или число	Версия формата
Width	Число	Ширина изображения в пикселах
Height	Число	Высота изображения в пикселах
BitDepth	Число	Глубина цвета изображения в битах на пиксел
ColorType	Строка	Тип изображения: 'truecolor' или 'RGB' – для полноцветных изображений; 'grayscale' – для полутоновых изображений; 'indexed' – для палитровых изображений

## 1.2 Вывод изображений на дисплей

Изображения можно вывести на дисплей с помощью функции `imshow`, которая имеет следующий синтаксис:

```
imshow(f, n)
```

Функция `imshow(f, n)` выводит на экран полутоновое изображение `f`, используя при выводе `n` уровней серого. Если при вызове функции опустить параметр `n`, то когда MATLAB запущен в графическом режиме `TrueColor`, для вывода полутонового изображения используется 256 градаций серого или 64 градации серого, когда MATLAB запущен в графическом режиме с меньшим количеством цветов.

```
imshow(f, [low high])
```

Функция `imshow(f, [low high])` выводит на экран полутоновое изображение `f`, дополнительно контрастируя выводимое изображение. Пикселы изображения `f`, яркость которых меньше либо равна `low`, отображаются черным цветом. Пикселы, яркость которых больше либо равна `high`, отображаются белым цветом. Пикселы, яркость которых имеет значение между `low` и `high`, отображаются серым цветом. Все уровни серого равномерно распределены от `low` до `high`. Если вызвать функцию

`imshow(f, [ ])`, указав вторым аргументом пустой массив, то `low` будет присвоено минимальное значение в `f(low = min(I(:)))`, а `high` будет присвоено максимальное значение в `f(max = max(I(:)))`.

Для вывода нескольких изображений в одном окне может быть использована функция `subimage` в сочетании с функцией `subplot`:

```
subplot(m, n, p), subimage(f);
```

Функция `subplot(m, n, p)` разбивает текущее окно на  $m \times n$  подокон и устанавливает текущим окно с номером `p`. Подокна нумеруются слева направо и сверху вниз, начиная от левого верхнего подокна, которое имеет номер 1. Функция `subimage` выводит изображение `f` в текущее подокно.

Другая функция приложения `imtool` служит для запуска инструмента Image Tool, которая представляет собой интегрированную среду для визуализации изображений и выполнения некоторых операций обработки и анализа изображений. Инструментарий Image Tool обеспечивает отображение изображений всех типов, которые поддерживаются системой MATLAB, а также доступ к другим средствам навигации и анализа изображений, например, к Pixel Region tool, Image Information tool, Adjust Contrast tool и др.

### 1.3 Сохранение изображений

Запись изображений на диск осуществляется функцией `imwrite`, которая имеет синтаксис:

```
imwrite(f, 'filename');
```

При такой записи строка `filename` должна содержать расширение, поддерживаемое средой MATLAB (таблица 1). Иначе желаемый формат файла можно задать с помощью третьего аргумента функции:

```
imwrite(f, 'filename', fmt);
```

Параметр `fmt` является строкой в которую записывается значение требуемого формата (`jpg`, `bmp`, `tiff` и др.).

Функция `imwrite` может иметь другие параметры, зависящие от выбранного формата. Для выполнения лабораторных работ мы будем иметь дело в основном с графическим форматом JPEG; в этом случае функция `imwrite` будет иметь вид:

```
imwrite(f, 'filename.jpg', 'quality', q);
```

где `q` – целое число в интервале от 0 до 100, определяющее степень искажения изображения при сжатии в формате JPEG.

Часто бывает необходимо сохранить изображение на диске точно в том виде, котором оно отображается на экране. Особенно важно уметь это

делать для графиков и диаграмм. Для экспорта изображений применяется команда `print`:

```
print -fno -dfileformat -rresno -filename;
```

где `no` – номер окна интересующего окна, `fileformat` – один из возможных форматов изображения (таблица 1), `resno` – разрешение в dpi, `filename` – имя, под которым будет сохранен файл изображения.

#### 1.4 Типы данных MATLAB и их конвертация

Несмотря на то, что цифровое изображение в большинстве случаев представляется целочисленными положительными значениями пикселей, в процессе обработки они могут быть преобразованы и в результате получиться как отрицательными, комплексными либо с плавающей точкой. В таблице 3 представлены основные типы данных, используемые в среде MATLAB.

Таблица 3 – Основные числовые типы данных MATLAB

Тип данных	Диапазон значений	Описание
<code>double</code>	-10 308...10 308 (8 байт на число)	Вещественные числа с плавающей запятой двойной точности
<code>uint8</code>	0...255 (1 байт на число)	Целые без знака
<code>uint16</code>	0...65 535 (2 байта на число)	
<code>uint32</code>	0...4 294 967 295 (4 байт на число)	
<code>int8</code>	-128...127 (1 байт на число)	Целые со знаком
<code>int16</code>	-32 768...32 767 (2 байта на число)	
<code>int32</code>	-2 147 483 648...2 147 483 647 (4 байта на число)	
<code>single</code>	-1 038...1 038 (4 байта на число)	Вещественные числа с плавающей запятой обычной точности
<code>char</code>	Буквы и знаки (2 байта на символ)	Символы
<code>logical</code>	0 и 1 (1 байт на элемент)	Логический тип

Все численные операции в MATLAB выполняются с двойной точностью в типе данных `double`. Другой часто встречающийся типом



данных является uint8, который возникает при считывании данных с запоминающих устройств. Эти два типа данных, а также тип logical и в меньшей степени uint16, образуют первостепенные типы данных.

Преобразование (конвертирование) изображений из одних типов данных в другие является самым распространенным действием в IPT. Совершая преобразование типов данных, необходимо помнить о диапазонах значений каждого типа данных (таблица 3).

В общем случае типы данных конвертируются следующим образом:

$$B = \text{data\_class\_name}(A);$$

где data\_class\_name – имя требуемого типа данных. Например, для преобразования массива с дробными числами типа double в тип uint8 (значения от 0 до 255), необходимо выполнить команду  $B = \text{uint8}(A)$ . Стоит отметить, что все отрицательные значения будут преобразованы в 0, а величины больше 255 – в 255, при этом у всех остальных элементов будет отброшена дробная часть. Это значит, что для правильного преобразования необходимо совершать предварительную перенормировку (масштабирование) его элементов.

В пакете IPT имеются специальные функции (таблица 4), которые совершают перенормировку при конвертации из одного типа данных в другие. Например, функция im2uint8 сначала распознает тип данных на входе и совершает все необходимые операции, чтобы выходное изображение имело правильный тип данных. Однако если на входе были отрицательные значения, то функция их обнуляет, на место значений, больших 1, ставит 255, все остальные умножает на 255 и округляет до ближайшего целого значения. Для правильной конвертации типа данных необходимо произвести перенормировку с использованием функции mat2gray, которая осуществляет преобразование значений в тип данных double в диапазоне [0...1]:

$$G = \text{mat2gray}(A, [\text{Amin} \text{Amax}]);$$

Изображение G приобретает значения в диапазоне от 0 до 1. Все элементы меньше Amin обнуляются, а все элементы больше Amax заменяются на 1. При выполнении команды  $g = \text{mat2gray}(A)$  значения Amin и Amax будут иметь значения максимума и минимума массива A.

Таблица 4 – Функции конвертации типов данных пакета IPT

Функция	Преобразует в	Допустимый тип данных
im2uint8	uint8	logical, uint8, uint16, double
im2uint16	uint16	logical, uint8, uint16, double
im2double	double	logical, uint8, uint16, double
im2bw	logical	uint8, uint16, double
mat2gray	double (область [0...1])	double

## 1.5 Операции с массивами

В среде MATLAB предусмотрено несколько мощных схем индексирования, которые существенно упрощают обращение с массивами и оптимизируют работу программы. Рассмотрим работу с массивами на примере одномерного массива размера  $1 \times N$  (вектор-строка). Элементы такого массива можно получить с помощью одномерного индексирования. Например,  $A(1)$  – это первый элемент вектора  $A$ ,  $A(2)$  – второй элемент и т.д. При инициализации элементы массива следует помещать в квадратные скобки, разделяя их пробелами или запятыми:

```
A = [1 3 5 7 9];
```

Для обращения к набору элементов используется оператор двоеточия:  $A(1:3)$  – вернет первые три значения вектора  $A$ . Аналогично можно получить любой диапазон, например, со второго по четвертый –  $A(2:4)$ , или с третьего до последнего –  $A(3:end)$ , где слово `end` означает последний элемент вектора. Индексировать можно не только последовательные элементы; например, при задании команды  $A(1:2:end)$  будут возвращены значения с первого элемента до последнего с шагом 2. Шаг может быть отрицательным –  $A(end:-2:1)$  – индексирование начинается с последнего элемента, уменьшается на каждом шаге на 2 и останавливается при достижении первого элемента.

Матрицы в MATLAB можно представить в виде последовательности векторов-строк, заключенных в квадратные скобки и разделенных точкой с запятой. Например, команда  $A = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$  инициализирует

```
1 2 3
A = 4 5 6 , размером 3x3.
```

```
7 8 9
```

Для выбора элемента матрицы необходимо задать два индекса: один обозначает номер строки, второй – номер столбца:  $A(2, 3)$  вернет элемент, расположенный во второй строке и в третьем столбце. Оператор двоеточия может быть использован для выбора последовательности элементов:  $A(:, 3)$  – вернет все значения третьего столбца. Команда  $A(1:2, 1:end)$  извлечет первые 2 строки.

Часто при построении алгоритмов полезно уметь строить простые массивы для верификации и тестирования разрабатываемых функций. Рассмотрим стандартные функции, генерирующие некоторые специальные массивы.

Функция `zeros(M, N)` – генерирует матрицу  $M \times N$  из нулей типа данных `double`.

Функция `ones(M, N)` – генерирует матрицу  $M \times N$  из единиц типа данных `double`.

Функция `true(M, N)` – генерирует логическую матрицу  $M \times N$  из единиц (истина).

Функция `false(M, N)` – генерирует логическую матрицу  $M \times N$  из нулей (ложь).

Функция `rand(M, N)` – генерирует матрицу  $M \times N$ , элементами которой являются нормально распределенные (гауссовы) случайные значения со средним 0 и с дисперсией 1.

## **Лабораторная работа №1: Основы цифрового представления оптических изображений**

Цель работы – изучение типов изображений, функций (средств) системы MATLAB, применяемых для работы с файлами изображений, и приобретение практических навыков их использования

### **Краткие теоретические сведения**

Цифровое изображение – массив данных, полученный путем дискретизации и квантования оригинала. Будучи закодированным с помощью особого алгоритма и записанным на носитель, этот массив данных становится файлом. Цифровое изображение представляет собой двумерную дискретную функцию  $f(x, y)$ , где  $x$  и  $y$  – координаты в пространстве. Значение функции  $f(x, y)$  – называется интенсивностью, яркостью или уровнем серого в точке  $(x, y)$ . Величины  $f, x, y$  принимают конечное число дискретных значений. Существуют следующие основные типы цифровых изображений: полноцветные, полутоновые, двоичные.

Полноцветные или просто цветные (truecolor, rgb) изображения – изображения, пикселы которого непосредственно хранят информацию об интенсивностях цветовых составляющих. Каждый пиксель описывается тремя значениями красной, синей и зеленой составляющей.

Полутоновое (изображения в градациях серого intensity, grayscale) – цифровое изображение, пикселы которого могут принимать одно из значений интенсивности какого-либо одного цвета в диапазоне от минимальной до максимальной интенсивности;

Двоичное (бинарное) – изображения, пикселы которого могут принимать только два значения: 0 и 1 (черный и белый цвет);

### **Порядок выполнения работы**

1. Подготовить свой портрет в цифровом формате (цветное изображение с разрешением не более 800×800 пикселей, формата .jpg или .png).
2. Произвести чтение исходного изображения в MATLAB.
3. Отобразить исходное изображение на экране.
4. Сохранить изображение в директории «.../DIP/Lab1» в формате jpg.
5. Сохранить изображение в директории «.../DIP/Lab1» в формате png.
6. Получить информацию о сохраненных файлах изображениях.
7. Рассчитать и сравнить степень сжатия изображений в форматах jpg и png по формуле:

$$K_s = \frac{(W \cdot H \cdot Bit) / 8}{FileSize},$$

где  $K_s$  – степень сжатия изображения;  $W, H$  – высота и ширина изображения в пикселях;  $Bit$  – разрядность изображения в бит;  $FileSize$  – размер файла изображения в байтах.

8. Преобразовать исходное изображение в полутоновое и вывести на экран. Сохранить изображение в директории «.../DIP/Lab1».
9. Преобразовать полутоновое изображение в двоичное с порогом 25, 50, 75% и вывести на экран. Сохранить изображения в директории «.../DIP/Lab1/Logical».
10. Преобразовать полутоновое изображение в 8-битовое, разбить на битовые плоскости. Результат преобразовать в двоичное изображение и вывести на экран каждую битовую плоскость. Сохранить каждую отдельную битовую плоскость в директорию «.../DIP/Lab1/BitPlane».
11. Выполнить дискретизацию полутонового изображения с ядром  $5 \times 5$ ,  $10 \times 10$ ,  $20 \times 20$ ,  $50 \times 50$ . Дискретизация реализуется функцией `blkproc(I, [m n], 'mean2(x)*ones(size(x))')`; Сохранить изображения в директории «.../DIP/Lab1/Discret».
12. Выполнить квантование полутонового изображения по уровню 4, 16, 32, 64, 128. Сохранить изображения в директории «.../DIP/Lab1/Quantiz».
13. Вырезать из центра полутонового изображения область размером  $100 \times 100$  пикселей и вывести на экран. Сохранить изображение вырезанной области в директории «.../DIP/Lab1/Crop».
14. Сохранить в массивы  $N_1, N_2, N_3$  значения четверки соседей пикселя  $p_1(21, 17)$ , четверки соседей по диагонали пикселя  $p_2(15, 11)$ , восьмерки соседей пикселя  $p_3(19, 88)$  соответственно.
15. Посчитать средний уровень яркости на изображении.
16. Нанести на полутоновое изображение метки в форме квадрата размером  $20 \times 20$  пикселей в центр и по углам изображения – если средний уровень яркости меньше 128 – белым цветом, если больше – черным. Вывести изображение на экран и сохранить в директории «.../DIP/Lab1/Marks».

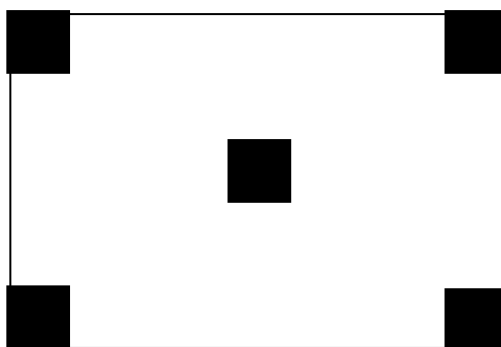


Рисунок 1 – Расположение марок на полутоновом изображении

В отчете привести код программы и полученные результаты выполнения программы.

#### **Вопросы для подготовки к защите**

1. Что такое цифровое изображение, пиксель, цифровая обработка изображений? На какие уровни условно можно разделить методы цифровой обработки изображений?
2. Предпосылки и история развития цифровой обработки изображений.
3. Назначение и области применения методов цифровой обработки изображений.
4. Этапы формирования изображения.
5. Квантование и дискретизация сигнала.
6. Битовые плоскости.
7. Типы изображений.
8. Соотношения между пикселями

## Лабораторная работа №2: Синтезирование изображений

Цель работы – приобретение навыков синтеза цифровых изображений, генерации шума различной природы возникновения. Использование базовых арифметических и геометрических операций над цифровым изображением.

### Краткие теоретические сведения

Под синтезированным цифровым изображением понимается любое визуальное представление информации, получаемое в результате вычислений в некоторой вычислительной системе (компьютере), выполняющей некоторые директивы пользователя (получателя изображения), вне всякой физической связи с реальным изображением. Последнее замечание исключает из предмета цифровой обработки изображений такие задачи, как обработка изображений, получаемых при фотографии или видеосъемке в реальном времени, поскольку в данном случае имеют дело с физическими объектами, а не с построением образов графических объектов по тем или иным их описаниям. Например, описанием окружности, изображение которой надо получить на экране монитора, может служить ее уравнение, или координаты центра и радиус, или двоичное представление ее контура.

Множество видимых на экране элементов, упорядоченных согласно определенным пространственным отношениям (в качестве таких отношений можно, например, назвать отношения «дальше-ближе», «больше-меньше» и т.д.), с заданными атрибутами окружения (например, уровень освещенности, признак наличия дымки и её плотности), с атрибутами поверхностей элементов (цвет, текстура, оптические свойства материалов моделируемых объектов) образуют сцену. Видимые глазом графические объекты, из которых состоит сцена, будем называть элементами сцены.

Одной из главных составляющих на изображении является шум. Шумом называется дефект изображения, вносимый влиянием окружающей среды, фотосенсорами и электроникой устройств, которые их используют, вследствие несовершенства технологий. Шум проявляется на изображении в виде наложенной маски из пикселей случайного цвета и яркости. Синтезирование шума на изображении принято реализовывать на основе заданной функции плотности вероятности или функции распределения. Большинство генераторов случайных чисел в языках программирования (включая MATLAB) основаны на переформулировании задачи в терминах случайных чисел с равномерной функцией распределения в интервале  $[0, 1]$  – с помощью функции `rand`. На основе равномерного закона распределения могут быть получены любые виды распределения шумов. В

основе этого подхода лежит одно из утверждений теории вероятности о том, что если имеется случайная величина с равномерным распределением на отрезке  $[0, 1]$ , то случайную величину  $z$  с заданной функцией распределения  $F_z$  можно построить по формуле:

$$z = F^{-1}(w),$$

т.е. в эквивалентной форме необходимо решить уравнение вида  $F(z) = w$  относительно  $z$ .

Например, для генерации случайной величины  $z$  с функцией распределения вероятностей Релея, следует воспользоваться следующей формулой:

$$F_z(z) = \begin{cases} 1 - e^{-(z-a)^2/b} & \text{при } z \geq a \\ 0 & \text{при } z < a \end{cases}$$

При использовании случайных чисел с равномерным распределением в интервале  $[0, 1]$ , чтобы получить  $z$ , достаточно решить уравнение:

$$1 - e^{-(z-a)^2/b} = w \quad \text{или} \quad z = a + \sqrt{b \ln(1-w)}$$

Поскольку квадратный корень – положительная функция, то генерируемые случайные величины будут всегда больше  $a$ , что требуется в определении функции Релея. Значит, равномерно распределенные случайные числа смогут служить основой для генератора релеевских случайных чисел с заданными параметрами  $a$  и  $b$ .

Выражение вида  $z = a + \sqrt{b \ln(1-w)}$  иногда называют уравнением генератора случайных чисел, поскольку в нем определяется, как вычислять требуемые случайные величины. В этом случае имеется простая формула для решения уравнения. Однако это не всегда возможно, и проблему можно сформулировать следующим образом: как получить уравнение генератора случайных чисел, выход которого хорошо приближает случайную величину с заданной функцией плотности вероятностей.

В таблице приведены формы представления случайной величины. Для некоторых из них удастся выписать формулу обратной функции распределения, например, для экспоненциального распределения или для функции распределения Релея. В этих случаях имеется простая формула для выражения требуемых случайных чисел в терминах равномерных случайных чисел. В других случаях, как в случае гауссовой или логарифмически нормальной плотности, такой простой формулы не существует. Тогда необходимо искать альтернативный способ – например, для случайной величины  $z$  с логарифмически нормальной плотностью можно воспользоваться тем фактом, что величина  $\ln(z)$  имеет гауссово



распределение, и выписать выражение в терминах гауссовой случайно величины с нулевым средним и единичной дисперсией.

Имя	Плотность	Среднее и дисперсия	Распределение	Генератор
Равно- мерная	$p_z(z) = \begin{cases} \frac{1}{b-a}, & a \leq z \leq b \\ 0, & \text{иначе} \end{cases}$	$m = \frac{a+b}{2},$ $\sigma^2 = \frac{(b-a)^2}{12}$	$F_z(z) = \begin{cases} 0, & z < a \\ \frac{z-a}{b-a}, & a \leq z \leq b \\ 0, & z > b \end{cases}$	rand(MATLAB)
Гауссова	$p_z(z) = \frac{1}{\sqrt{2\pi b}} e^{-(z-a)^2/2b^2},$ $-\infty < z < +\infty$	$m = a,$ $\sigma^2 = b^2$	$F_z(z) = \int_{-\infty}^z p_z(v) dv$	randn(MATLAB)
«Соль и перец»	$p_z(z) = \begin{cases} P_a, & z = a \\ P_b, & z = a, (b > a) \\ 0, & \text{иначе} \end{cases}$	$m = aP_a + bP_b,$ $\sigma^2 = (a-m)^2 P_a + (b-m)^2 P_b$	$F_z(z) = \begin{cases} 0, & z < a \\ P_a, & a \leq z < b \\ P_a + P_b, & z \geq b \end{cases}$	rand(MATLAB)
Логарифм ически норма- льная	$p_z(z) = \frac{1}{\sqrt{2\pi bz}} e^{-(z-a)^2/2bz}, \quad z > 0$	$m = e^{a+(b^2/2)},$ $\sigma^2 = [e^{b^2} - 1] e^{2a+b^2}$	$F_z(z) = \int_{-\infty}^z p_z(v) dv$	$z = ae^{bN(0,1)}$
Релея	$p_z(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b}, & z \geq a \\ 0, & z < a \end{cases}$	$m = a + \sqrt{2\pi b/4},$ $\sigma^2 = \frac{b(4-\pi)}{4}$	$F_z(z) = \begin{cases} 1 - e^{-(z-a)^2/b}, & z \geq a \\ 0, & z < a \end{cases}$	$z = a + \sqrt{b \ln[1 - U(0,1)]}$
Экспонен- циальная	$p_z(z) = \begin{cases} ae^{-az}, & z \geq 0 \\ 0, & z < 0 \end{cases}$	$m = \frac{1}{a},$ $\sigma^2 = \frac{1}{a^2}$	$F_z(z) = \begin{cases} 1 - e^{-az}, & z \geq 0 \\ 0, & z < 0 \end{cases}$	$z = -\frac{1}{a} \ln[1 - U(0,1)]$
Эрл анга	$p_z(z) = \frac{a^b z^{b-1}}{(b-1)!} e^{-az}, \quad z \geq 0$	$m = \frac{b}{a},$ $\sigma^2 = \frac{b}{a^2}$	$F_z(z) = \left[ 1 - e^{-az} \sum_{n=0}^{b-1} \frac{(az)^n}{n!} \right],$	$z = E_1 + E_2 + \dots + E_b,$ $E_n - \text{эксп. величины с}$ параметром $a$

В таблице  $N(0,1)$  – нормальная (гауссова) случайная величина со средним ноль и дисперсией 1.  $U(0,1)$  – равномерная случайная величина из интервала (0, 1).

Представленные в таблице генераторы случайной величины играют важную роль при моделировании поведения случайного шума в приложениях обработки изображений. Так гауссов шум используется в качестве естественного приближения в тех случаях, когда детекторы изображения работают на пороге чувствительности. Шум типа «соль и перец» возникает в устройствах с ошибочной коммутацией. Размеры зерен на фотоэмульсии являются случайными величинами с логарифмически нормальным законом распределения. Шум Релея образуется при фиксации удаленных изображений, а экспоненциальный шум и шум Эрланга используются при описании искажений на изображениях, полученных с помощью лазерного излучения.

### **Пример кода функции для генерации различных шумов на изображении**

```
function R = imnoise2(type, M, N, a, b)
%IMNOISE2 Generates an array of random numbers with specified PDF.
% R = IMNOISE2(TYPE, M, N, A, B) generates an array, R, of size
% M-by-N, whose elements are random numbers of the specified TYPE
% with parameters A and B. If only TYPE is included in the
% input argument list, a single random number of the specified
% TYPE and default parameters shown below is generated. If only
% TYPE, M, and N are provided, the default parameters shown below
% are used. If M = N = 1, IMNOISE2 generates a single random
% number of the specified TYPE and parameters A and B.
%
% Valid values for TYPE and parameters A and B are:
%
% 'uniform'      Uniform random numbers in the interval (A, B).
%                The default values are (0, 1).
% 'gaussian'     Gaussian random numbers with mean A and standard
%                deviation B. The default values are A = 0, B = 1.
% 'salt & pepper' Salt and pepper numbers of amplitude 0 with
%                probability Pa = A, and amplitude 1 with
%                probability Pb = B. The default values are Pa =
%                Pb = A = B = 0.05. Note that the noise has
%                values 0 (with probability Pa = A) and 1 (with
%                probability Pb = B), so scaling is necessary if
%                values other than 0 and 1 are required. The noise
%                matrix R is assigned three values. If R(x, y) =
%                0, the noise at (x, y) is pepper (black). If
%                R(x, y) = 1, the noise at (x, y) is salt
%                (white). If R(x, y) = 0.5, there is no noise
%                assigned to coordinates (x, y).
% 'lognormal'    Lognormal numbers with offset A and shape
%                parameter B. The defaults are A = 1 and B =
%                0.25.
% 'rayleigh'     Rayleigh noise with parameters A and B. The
%                default values are A = 0 and B = 1.
% 'exponential' Exponential random numbers with parameter A. The
```

```

%           default is A = 1.
%   'erlang'   Erlang (gamma) random numbers with parameters A
%             and B.  B must be a positive integer.  The
%             defaults are A = 2 and B = 5.  Erlang random
%             numbers are approximated as the sum of B
%             exponential random numbers.

%   Copyright 2002-2006 R. C. Gonzalez, R. E. Woods, & S. L. Eddins
%   Digital Image Processing Using MATLAB, Prentice-Hall, 2004
%   $Revision: 1.6 $   $Date: 2006/07/15 20:44:52 $

% Set default values.
if nargin == 1
    a = 0; b = 1;
    M = 1; N = 1;
elseif nargin == 3
    a = 0; b = 1;
end

% Begin processing.  Use lower(type) to protect against input being
% capitalized.
switch lower(type)
case 'uniform'
    R = a + (b - a)*rand(M, N);
case 'gaussian'
    R = a + b*randn(M, N);
case 'salt & pepper'
    if nargin <= 3
        a = 0.05; b = 0.05;
    end
    % Check to make sure that Pa + Pb is not > 1.
    if (a + b) > 1
        error('The sum Pa + Pb must not exceed 1.')
    end
    R(1:M, 1:N) = 0.5;
    % Generate an M-by-N array of uniformly-distributed random numbers
    % in the range (0, 1).  Then, Pa*(M*N) of them will have values <=
    % a.  The coordinates of these points we call 0 (pepper
    % noise).  Similarly, Pb*(M*N) points will have values in the range
    % > a & <= (a + b).  These we call 1 (salt noise).
    X = rand(M, N);
    c = find(X <= a);
    R(c) = 0;
    u = a + b;
    c = find(X > a & X <= u);
    R(c) = 1;
case 'lognormal'
    if nargin <= 3
        a = 1; b = 0.25;
    end
    R = exp(b*randn(M, N) + a);
case 'rayleigh'
    R = a + (-b*log(1 - rand(M, N))).^0.5;
case 'exponential'
    if nargin <= 3
        a = 1;
    end
    if a <= 0
        error('Parameter a must be positive for exponential type.')
    end
    k = -1/a;
end

```

```

R = k*log(1 - rand(M, N));
case 'erlang'
    if nargin <= 3
        a = 2; b = 5;
    end
    if (b ~= round(b) | b <= 0)
        error('Param b must be a positive integer for Erlang.')
    end
    k = -1/a;
    R = zeros(M, N);
    for j = 1:b
        R = R + k*log(1 - rand(M, N));
    end
otherwise
    error('Unknown distribution type.')
end

```

### **Порядок выполнения работы**

1. Синтезировать пустое полутоновое 8-битное изображение размером 800×800 пикселей.
2. Нанести на изображение шум в соответствии с вариантом индивидуального задания по известному закону распределения.
3. Построить гистограмму распределения полученного изображения шума. Сохранить график гистограммы в директорию «...DIP\Lab2».
4. Синтезировать изображение Объекта 1 в соответствии с вариантом индивидуального задания в центре изображения (см. Рисунок 2). Сохранить изображение в директорию «...DIP\Lab2».
5. Выполнить масштабирование изображения, увеличив и уменьшив размер синтезированного объекта в 2 раза разными методами (ближайшего соседа, билинейной или бикубической интерполяции). Сохранить изображения в директорию «...DIP\Lab2».
6. Синтезировать новое изображение с шумом и нанести объекты 2 и 3 в соответствии с вариантом индивидуального задания. Объекты должны располагаться в верхнем левом и нижнем правом углах синтезированного изображения (см. Рисунок 2). Размеры объектов подобрать самостоятельно, исходя из того, что наибольшая сторона (или диаметр) должна быть не менее 80 пикселей.
7. Выполнить операцию зеркального отражения по горизонтали. Сохранить изображение в директорию «...DIP\Lab2».
8. Выполнить операцию зеркального отражения по вертикали. Сохранить изображение в директорию «...DIP\Lab2».
9. Выполнить операцию поворота изображения по часовой стрелке на 45°. Сохранить изображение в директорию «...DIP\Lab2».
10. Выполнить операцию поворота изображения против часовой стрелки на 45°. Сохранить изображение в директорию «...DIP\Lab2»

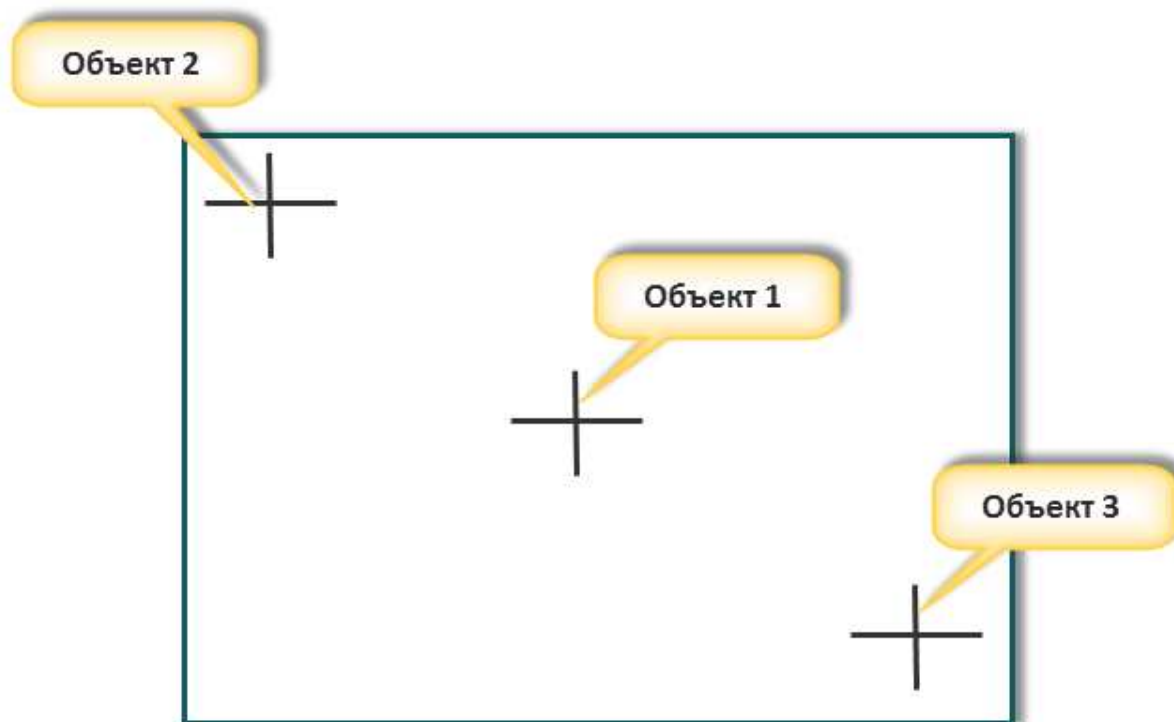




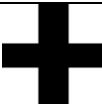










































Рисунок 2 – Расположение синтезированных объектов на изображении

11. Выбрать из папки вспомогательных материалов «...\Labs\Lab2\Fon» любое из предложенных изображений фона.
12. Вырезать из изображения фона любой участок размером 800×800 пикселей.
13. Уменьшить яркость вырезанного фона в 4 раза.
14. Сформировать новое полутоновое изображение из вырезанного фона с уменьшенной яркостью, 2-х объектов и добавить на изображение шум в соответствии с вариантом индивидуального задания. Сохранить полученное изображение в директорию «...DIP\Lab2».
15. Выполнить операцию «Негатив» для синтезированного изображения. Сохранить полученное изображение в директорию «...DIP\Lab2».
16. Сформировать новое полутоновое изображение из вырезанного фона с уменьшенной яркостью, одного объекта и добавить на изображение шум в соответствии с вариантом индивидуального задания. Сохранить полученное изображение в директорию «...DIP\Lab2».
17. Выполнить операцию разности двух изображений, полученных в пп. 14 и 16. Сохранить полученное изображение в директорию «...DIP\Lab2».

### Варианты индивидуального задания

*Примечание.* Синтезированный объект должен быть белого цвета. Размеры объекта выбираются таким образом, чтобы он был различим на изображении.

№	Объект 1	Объект 2	Объект 3	Шум
1.				Рэля
2.				«Соль и перец»
3.				Экспоненциальный
4.				Эрланга
5.				Логарифмически нормальный
6.				Равномерный
7.				Гауссов
8.				«Соль и перец»
9.				Гауссов
10.				Экспоненциальный

11.				«Соль и перец»
12.				Рэля
13.				Равномерный
14.				Эрланга
15.				Экспоненциальный

### Вопросы для подготовки к защите

1. Простейшие геометрические преобразования изображения.
2. Аффинные преобразования изображения.
3. Простейшие операции над изображением (арифметические операции).  
Линейные и нелинейные операции.
4. Шумы на изображении. Виды шума и источники шума на изображении.
5. Гистограмма изображения и ее свойства.



## **Лабораторная работа №3: Основы методов пространственной обработки изображений**

Цель работы – изучение основных методов пространственной обработки цифровых изображений. Приобретение навыков использования градационных преобразований, пространственной фильтрации, операций над гистограммой изображения.

### **Краткие теоретические сведения**

Яркостные градационные преобразования относятся к числу простейших методов улучшения изображений. Преобразование реализуется на основе выражения вида:

$$s = T(r),$$

Где  $r$  – яркость пикселя изображения на входе,  $s$  – яркость пикселя на выходе,  $T$  – преобразование, отображающее значение пикселя  $r$  в значение пикселя  $s$ .

Базовыми градационными преобразованиями, часто используемые для улучшения изображения, являются операции линейного преобразования (негатив, тождественное преобразование), логарифмическое (логарифм и обратный логарифм) и степенное ( $n$ -я степень и корень  $n$ -ой степени).

Подходом, дополняющим базовые методы градационного преобразования, является использование кусочно-линейных функций. Главное их преимущество состоит в том, что их форма может быть сколь угодно сложной. Основной недостаток кусочно-линейных функций заключается в том, что для их описания необходимо задавать значительно большее количество параметров.

Другим методом пространственной обработки изображения является пространственная фильтрация. Для выполнения линейной пространственной фильтрации используют метод двумерной пространственной свертки локальной окрестности обрабатываемого элемента с линейным оператором, которая называется маской или матрицей коэффициентов фильтра.

Алгоритм свертки заключается в том, что маска построчно сканирует исходное изображение. На каждом шаге находится сумма произведений элементов маски и соответствующих элементов исходного изображения, и найденное значение присваивается одному элементу результирующего изображения. Достигнув таким образом конца строки, маска смещается на одну строку вниз, в начало строки, и процесс повторяется. Имеется две тесно связанные концепции, которые необходимо понимать при выполнении линейной пространственной фильтрации. Первая – это корреляция, а вторая – свертка. Корреляция состоит в прохождении маски по изображению. С точки зрения механики процесса, свертка делается так же, но маску надо повернуть на  $180^\circ$  перед прохождением по изображению.

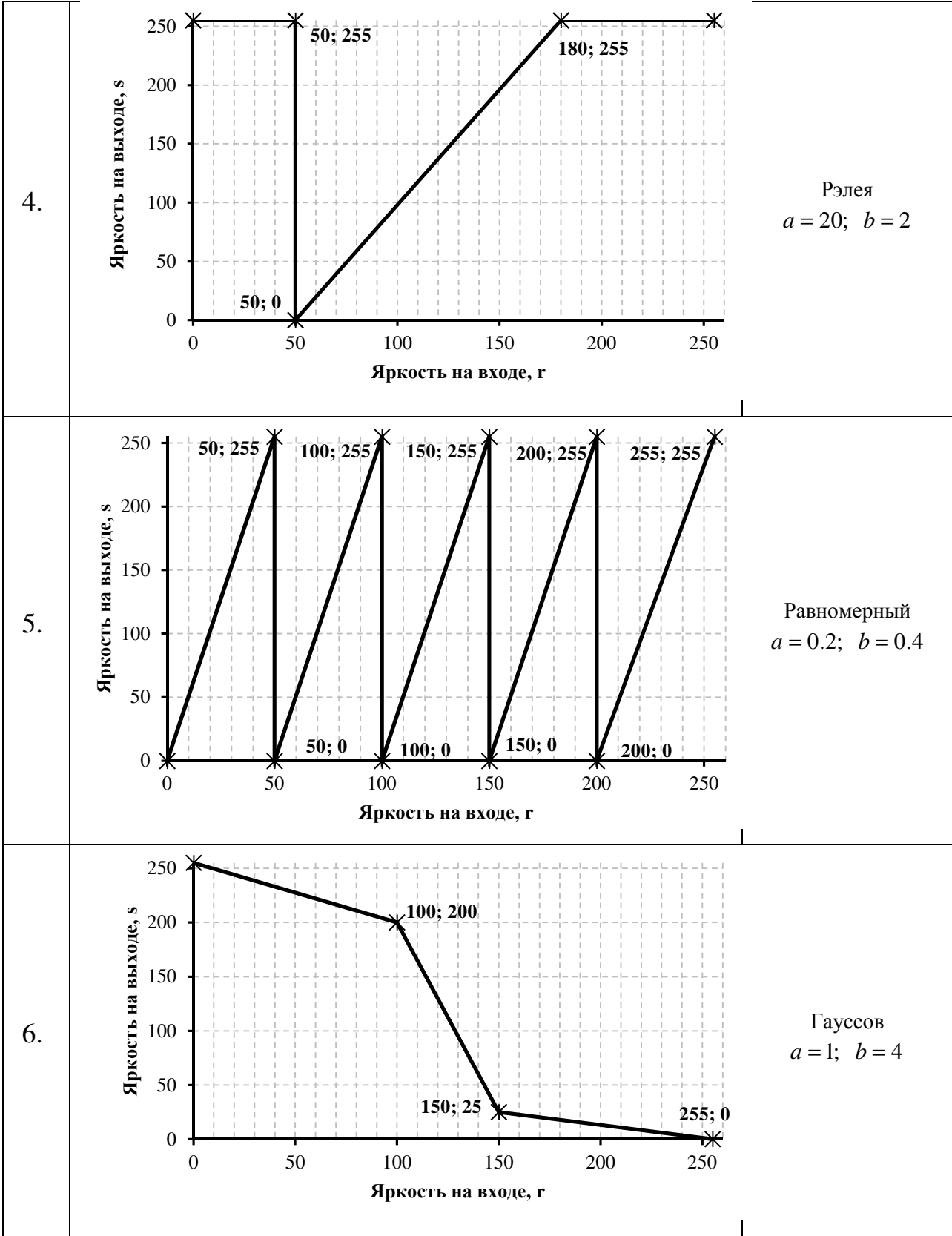
Если сдвигаемая маска является симметричной, то корреляция и свертка дают одинаковые результаты.

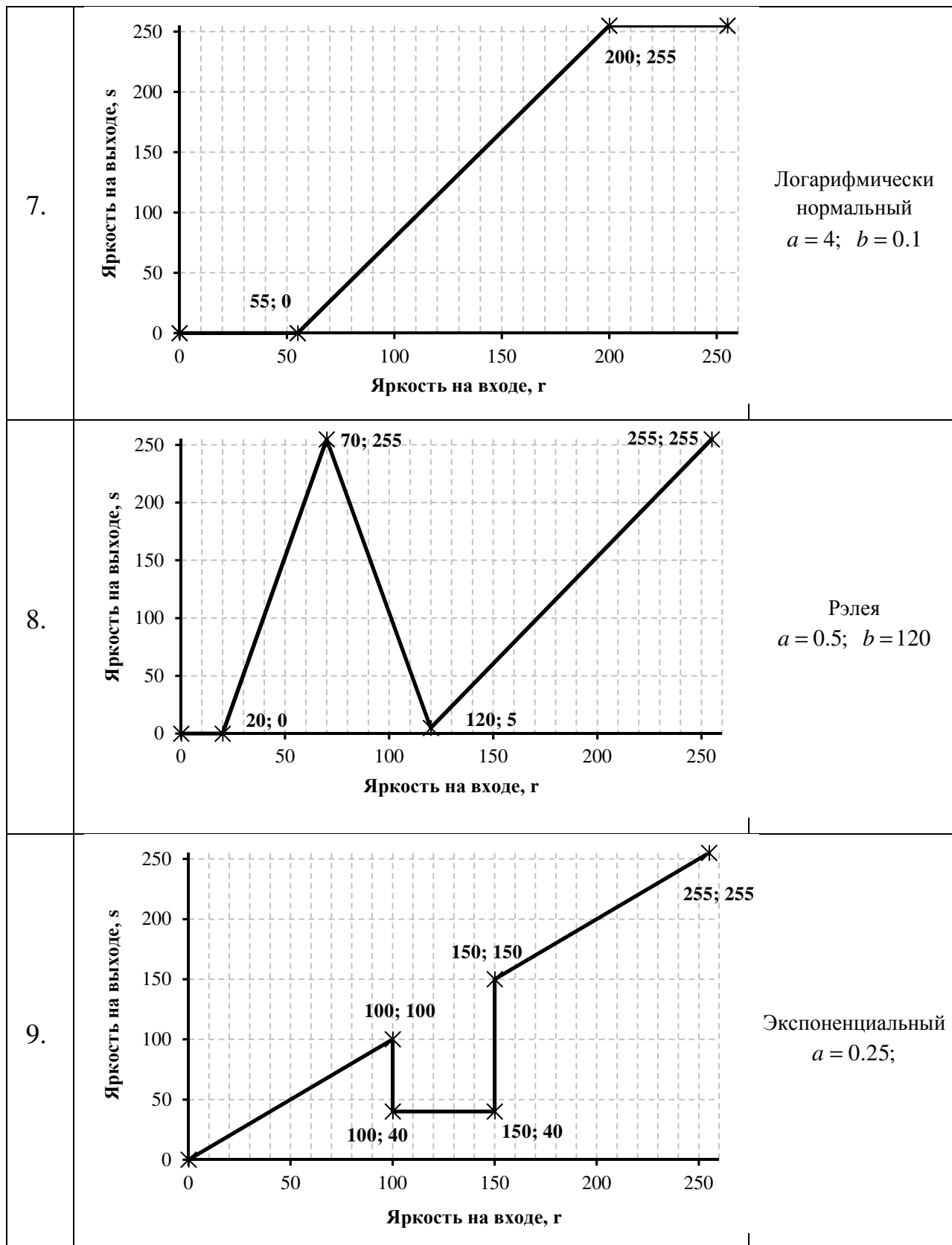
### **Порядок выполнения работы**

1. Подготовить свой портрет в цифровом формате (полутоновое изображение с разрешением не более  $800 \times 800$ ). Сохранить исходное изображение и его гистограмму в директорию «...DIP\Lab3».
2. Выполнить логарифмическое преобразование исходного изображения. Сохранить полученное изображение и его гистограмму в директорию «...DIP\Lab3\Log».
3. Выполнить степенное преобразование исходного изображения для значений  $\gamma = 0.1, 0.45, 5$ . Сохранить полученные изображения и их гистограммы при каждом значении  $\gamma$  в директорию «...DIP\Lab3\Degree».
4. Выполнить операцию кусочно-линейного преобразования в соответствии с вариантом индивидуального задания. Сохранить полученное изображение и его гистограмму в директорию «...DIP\Lab3\Line\_Contrast».
5. Выполнить операцию эквализации гистограммы над исходным изображением. Сохранить полученное изображение и его гистограмму в директорию «...DIP\Lab3\Equaliz».
6. Выполнить пространственную фильтрацию исходного изображения усредняющим фильтром с квадратной маской размерами 3, 15, 35. Сохранить полученные изображения в директорию «...DIP\Lab3\Filter».
7. Выполнить пространственную фильтрацию исходного изображения фильтром повышения резкости изображения. Сохранить полученное изображение в директорию «...DIP\Lab3\Filter».
8. Выполнить пространственную фильтрацию исходного изображения медианным фильтром с квадратной маской размерами 3, 9, 15. Сохранить полученное изображение в директорию «...DIP\Lab3\Median».
9. Выполнить операцию выделения границ исходного изображения методами Робертса, Превитта, Собеля. Сохранить полученные изображения в директорию «...DIP\Lab3\Edge».
10. Добавить на исходное изображение шум в соответствии с вариантом индивидуального задания. Подобрать фильтр и выполнить пространственную фильтрацию зашумленного изображения таким образом, чтобы дисперсия отфильтрованного изображения была близка к дисперсии исходного изображения. Сохранить зашумленное и фильтрованные изображения в директорию «...DIP\Lab3\Filter».
11. В отчете привести код программы и полученные результаты выполнения программы.

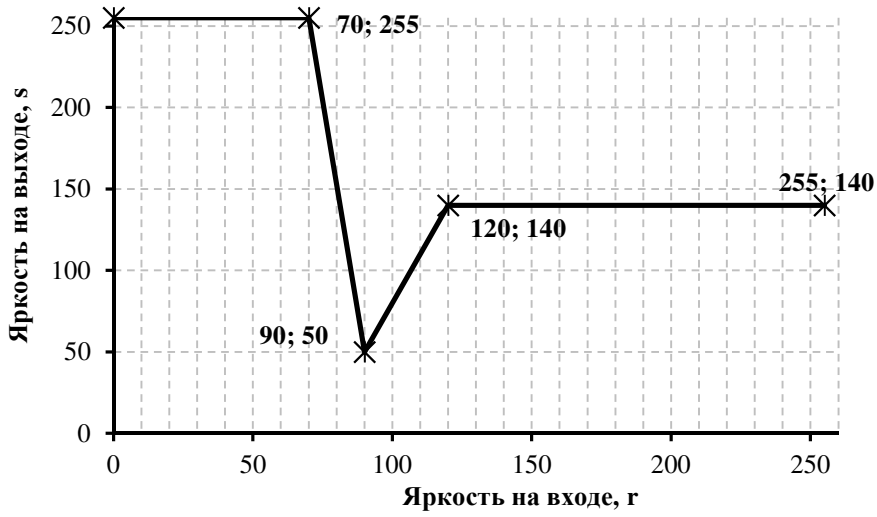
### Варианты индивидуального задания

№	Функция кусочно-линейного преобразования	Шум
1.	<p>Яркость на выходе, <math>s</math></p> <p>Яркость на входе, <math>г</math></p> <p>Points: (0; 0), (25; 10), (45; 180), (250; 250)</p>	<p>Экспоненциальный <math>a = 2.75;</math></p>
2.	<p>Яркость на выходе, <math>s</math></p> <p>Яркость на входе, <math>г</math></p> <p>Points: (0; 0), (65; 5), (65; 220), (120; 220), (120; 120), (250; 250)</p>	<p>Эрланга <math>a = 12; b = 20</math></p>
3.	<p>Яркость на выходе, <math>s</math></p> <p>Яркость на входе, <math>г</math></p> <p>Points: (0; 250), (120; 120), (120; 250), (180; 250), (180; 75), (250; 0)</p>	<p>«Соль и перец» <math>a = 0.1; b = 0.8</math></p>



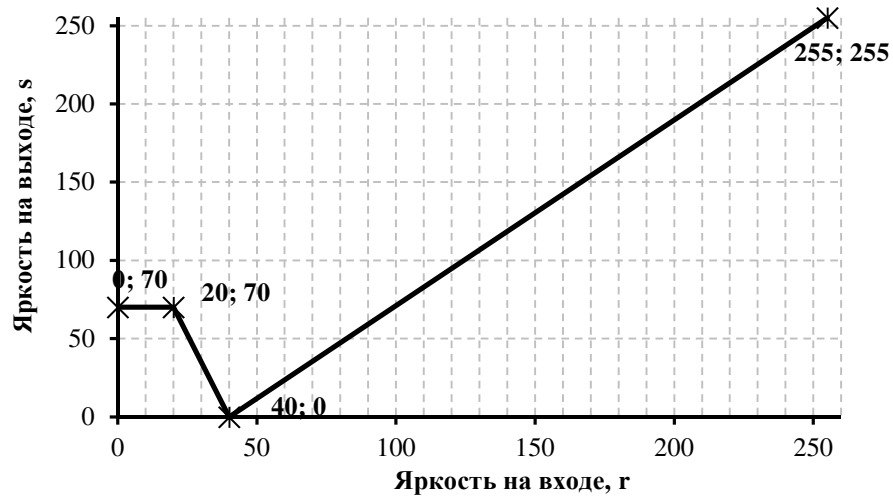


10.



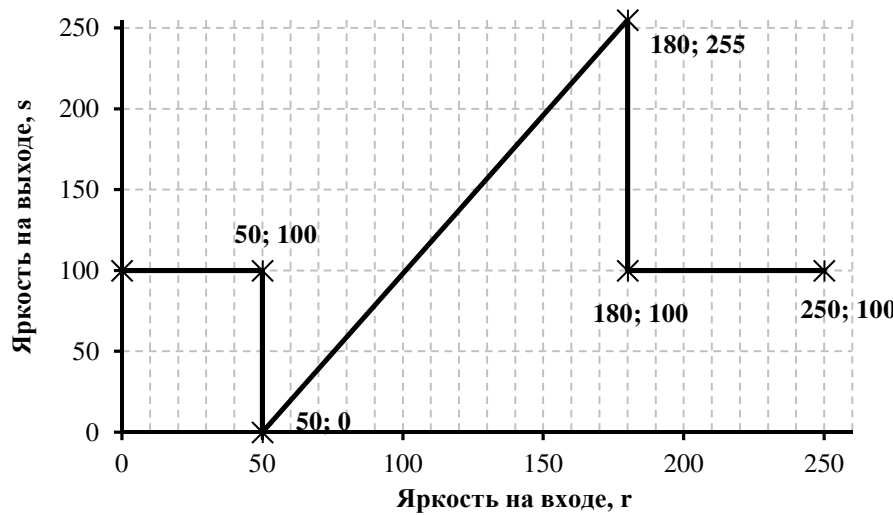
Гауссов  
 $a = 120; b = 15$

11.



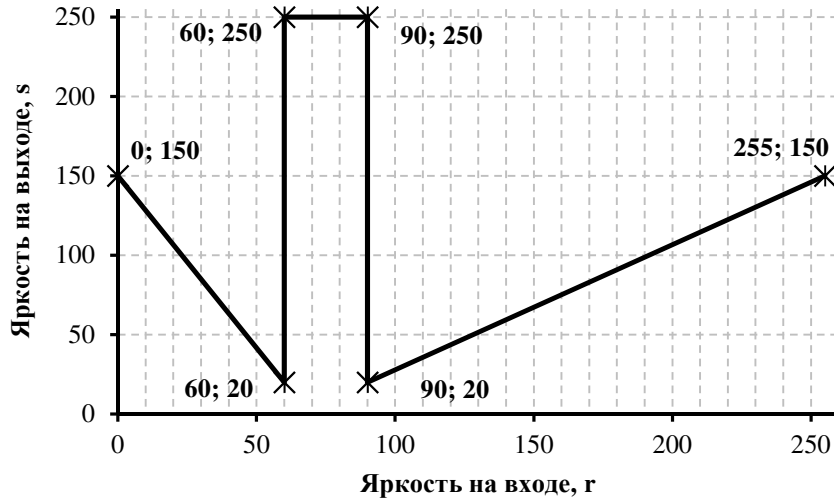
Эрланга  
 $a = 2; b = 2$

12.



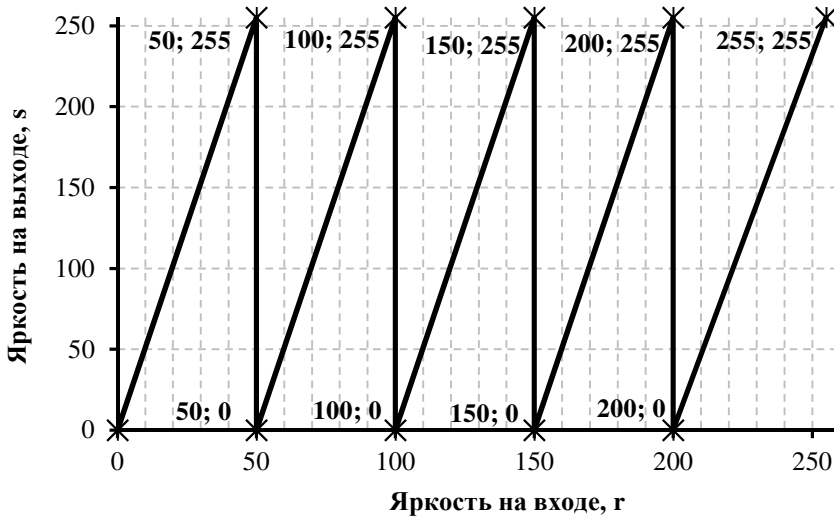
Логарифмически  
 нормальный  
 $a = 2; b = 0.55$

13.



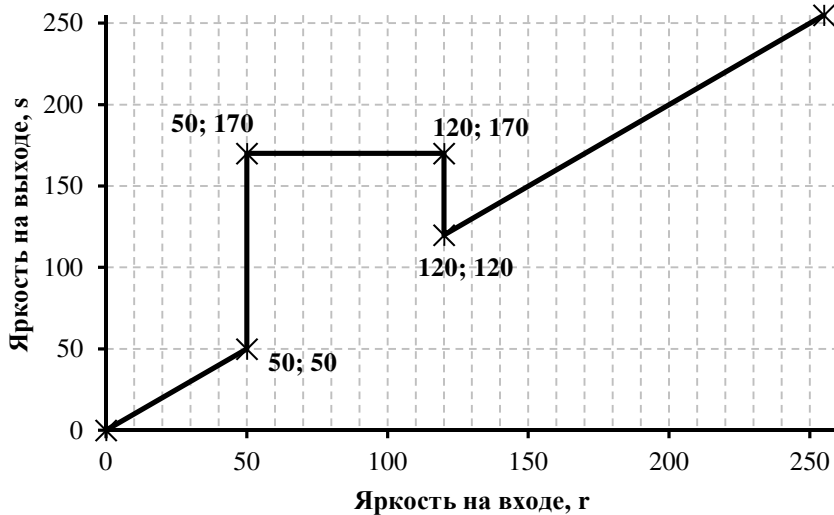
Экспоненциальный  
 $a = 10;$

14.



«Соль и перец»  
 $a = 0.1; b = 0.5$

15.



Равномерный  
 $a = 15; b = 1.5$

### **Вопросы для подготовки к защите**

1. Метода градационного преобразования изображения.
2. Методы пространственной фильтрации изображения.
3. Операции над гистограммой изображения.
4. Сглаживающие фильтры.
5. Фильтры повышения резкости изображения.
6. Фильтры выделения границ.



## Лабораторная работа №4: Фильтрация изображений в частотной области

Цель работы – изучение методов обработки цифровых изображений в частотной области. Приобретение навыков использования преобразования Фурье, частотной фильтрации.

### Краткие теоретические сведения

Линейная фильтрация изображений может осуществляться как в пространственной, так и в частотной области. При этом считается, что «низким» частотам соответствует основное содержание изображения – фон и крупноразмерные объекты, а «высоким» частотам – мелкоразмерные объекты, мелкие детали крупных форм и шумовая компонента.

Традиционно для перехода в область пространственных частот используются методы, основанные на преобразовании Фурье. Преобразование Фурье позволяет представить практически любую функцию или набор данных в виде комбинации таких тригонометрических функций, как синус и косинус, что позволяет выявить периодические компоненты в данных и оценить их вклад в структуру исходных данных или форму функции. Выделяют три основные формы преобразования Фурье: интегральное преобразование Фурье, ряды Фурье и дискретное преобразование Фурье. Основное отличие между тремя формами преобразования Фурье заключается в том, что если интегральное преобразование Фурье определено по всей области определения функции, то ряд и дискретное преобразование Фурье определены только на дискретном множестве точек, бесконечном для ряда Фурье и конечном для дискретного преобразования. Как видно из определений преобразования Фурье, наибольший интерес для систем цифровой обработки сигналов представляет дискретное преобразование Фурье. Данные, получаемые с цифровых носителей или источников информации, представляют собой упорядоченные наборы чисел, записанные в виде векторов или матриц.

Использование Фурье–спектра изображения для выполнения операций фильтрации обусловлено, прежде всего, высокой производительностью таких операций. Как правило, выполнение прямого и обратного двумерного преобразования Фурье и умножение на коэффициенты Фурье–спектра фильтра занимает меньше времени, чем выполнение двумерной свертки исходного изображения.

Алгоритмы фильтрации в частотной области основываются на теореме о свертке. В двумерном случае преобразование свертки выглядит следующим образом:

$$G(u, v) = F(u, v)H(u, v),$$

где  $G(u, v)$  – Фурье-спектр результата свертки (отфильтрованного изображения),  $H(u, v)$  – Фурье-спектр фильтра,  $F(u, v)$  – Фурье образ исходного изображения. То есть в частотной области двумерная свертка заменяется поэлементным перемножением образов исходного изображения и соответствующего фильтра.

### Порядок выполнения работы

1. Подготовить свой портрет в цифровом формате (полутоновое изображение с разрешением не более  $800 \times 800$ ). Получить изображение фурье-спектра. Сохранить исходное изображение и его фурье-спектр в директории «.../ДІР/Лаb4».

2. Провести низкочастотную фильтрацию изображения с использованием:

– идеального фильтра низких частот:

$$H(u, v) = \begin{cases} 1, & D(u, v) \leq D_0 \\ 0, & D(u, v) > D_0 \end{cases},$$

для  $D_0 = 5, 10, 50, 250$ .

– фильтра Баттерворта низких частот:

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}},$$

для  $n = 2$ ; и  $D_0 = 5, 10, 50, 250$

– фильтра Гауса низких частот:

$$H(u, v) = e^{-D^2(u, v) / 2D_0^2},$$

$D_0 = 5, 10, 50, 250$

3. Провести высокочастотную фильтрацию изображения с использованием:

– идеального фильтра высоких частот:

$$H(u, v) = \begin{cases} 0, & D(u, v) \leq D_0 \\ 1, & D(u, v) > D_0 \end{cases},$$

для  $D_0 = 5, 10, 50, 250$ .

– фильтра Баттерворта высоких частот:

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}},$$

для  $n = 2$ ; и  $D_0 = 5, 10, 50, 250$

– фильтра Гауса высоких частот:

$$H(u, v) = 1 - e^{-D^2(u, v) / 2D_0^2},$$

$D_0 = 5, 10, 50, 250$

4. Подобрать и провести избирательную фильтрацию изображения, хранящегося в файле «lab4\_var\_X.jpg», где X – номер индивидуального задания.
5. Результаты работы сохранять в директории: «.../DIP/Lab4/Low/...Ideal /... Butter/ ... Gaus» – для низкочастотной фильтрации «.../DIP/Lab4/High/ ...Ideal /... Butter/ ... Gaus» – для высокочастотной фильтрации

### **Вопросы для подготовки к защите**

1. Преобразование Фурье непрерывной и дискретной функции.
2. Свойства преобразования Фурье.
3. Основные соотношения преобразования Фурье.
4. Фильтрация в частотной области.
5. Низкочастотные фильтры.
6. Высокочастотные фильтры.

## **Лабораторная работа №5: Морфологическая обработка изображений**

**Цель работы** – изучение методов морфологической обработки изображений.

### **Краткие теоретические сведения**

Операции математической морфологии позволяют осуществлять обработку изображений с учетом формы и размера имеющихся на изображении областей. Морфологические операторы позволяют: выделять или удалять на изображениях мелко- и среднеразмерные объекты заданной формы и размера, а также фильтровать (сглаживать) форму крупноразмерных объектов.

Одним из основных понятий математической морфологии является понятие структурообразующего, или структурного элемента. Структурный элемент  $B$  – это множество, состоящее из двух непересекающихся подмножеств  $B_1$  и  $B_2$ , для которых определено общее начало.



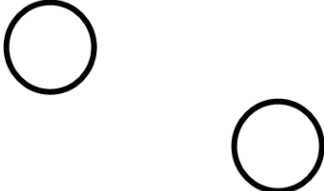

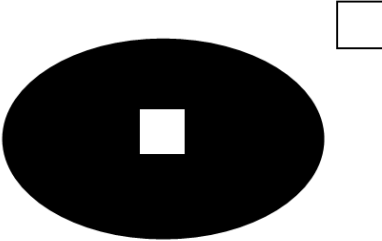
Морфологические операции дилатации и эрозии имеют основополагающее значение при морфологической обработке изображений. Операция эрозии «ужимает» или «утончает» объекты двоичных изображений. Операция дилатации «наращивает» или «утолщает» объекты на двоичных изображениях. Способ и степень этих преобразований контролируется формой структурного элемента. Эрозия и дилатация – операции, предназначенные в первую очередь для выявления различных морфологических особенностей изображения с использованием различных структурных элементов. Так, эрозия посредством круга с радиусом  $r$  позволяет найти в изображении объекты, минимальный поперечный размер которых превышает  $2r$ . Если в качестве структурного элемента взять две точки, смещение между которыми определяется вектором  $h$ , эрозия позволит выделить объекты, имеющие соседей в направлении и на расстоянии, заданных этим вектором.

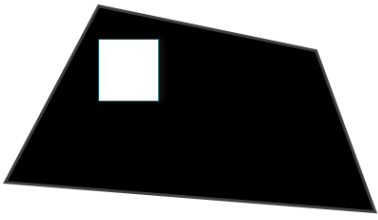
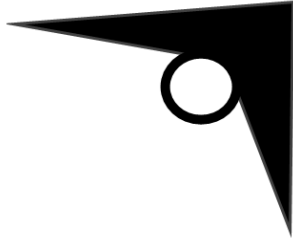
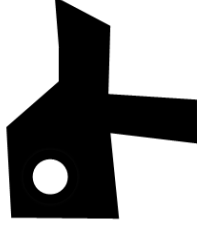

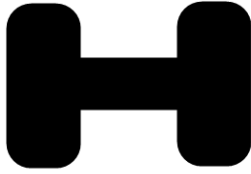
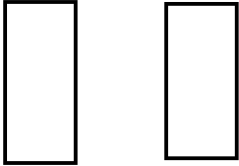
### **Порядок выполнения работы**

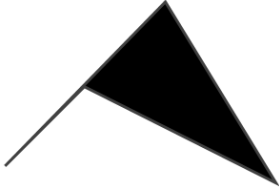

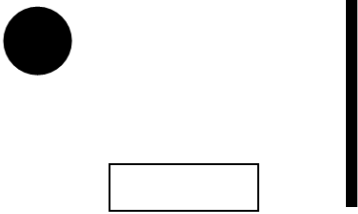
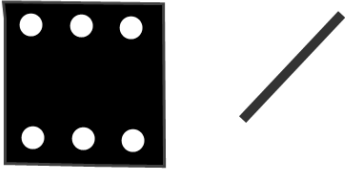
1. Загрузить исходное изображение в соответствии с вариантом индивидуального задания.
2. Выполнить необходимые преобразования над изображением с целью его улучшения.
3. Выполнить бинаризацию изображения.
4. Произвести морфологическую обработку изображению, чтобы в результате получилось изображение, близкое к варианту индивидуального задания.

5. Результаты работы сохранять в директории: «.../DIP/Lab5/».

**Варианты индивидуального задания**

№	Файл изображения	Выходное изображение	
1.	Lab5_Var1.png		
2.	Lab5_Var2.png		
3.	Lab5_Var3.png		
4.	Lab5_Var4.png		
5.	Lab5_Var5.png		

6.	Lab5_Var6.png			
7.	Lab5_Var7.png			
8.	Lab5_Var8.png			
9.	Lab5_Var9.png			
10.	Lab5_Var10.png			
11.	Lab5_Var11.png			

12.	Lab5_Var12.png			
13.	Lab5_Var13.png			
14.	Lab5_Var14.png			
15.	Lab5_Var15.png			

### Вопросы для подготовки к защите

1. Морфологический анализ. Базовые операции.
2. Морфологическая операция эрозия. Определение. Свойства.
3. Морфологическая операция дилатация. Определение. Свойства.
4. Морфологическая операция замыкания. Определение. Свойства
5. Морфологическая операция размыкания. Определение. Свойства

## **Лабораторная работа №6: Обработка и сегментация цветных изображений**

Цель работы – изучение алгоритмов цветовой коррекции и фильтрации изображений и исследование методов сегментации и описания объектов на цифровом изображении.

### **Краткие теоретические сведения**

Цветные цифровые изображения представляют собой совокупность трех цветовых плоскостей, каждая из которых характеризует одну независимую составляющую цвета, представленную в том же формате, что и обычное 8-битное полутоновое изображение. Следовательно, все процедуры обработки полутоновых изображений в яркостной области могут быть обобщены и на случай обработки цветных изображений. Специфика же здесь связана, прежде всего, с различными цветовыми моделями, позволяющими по-разному работать с разными цветовыми и другими составляющими изображения.

Назначение цветовой модели – дать средства описания цвета в пределах некоторого цветового охвата. Наиболее часто в компьютерном зрении используются модели RGB, CMY, YUV, YCbCr, HSV.

Сегментацией называется процесс подразделения сцены на составляющие части или объекты. Сегментация является одним из основных элементов работы автоматизированной системы технического зрения, так как именно на этой стадии обработки объекты выделяются из сцены для дальнейшего распознавания и анализа. Алгоритмы сегментации, как правило, основываются на двух фундаментальных принципах: разрывности и подобии. В первом случае основной подход основывается на определении контуров, а во втором – на определении порогового уровня и расширении области. Эти понятия применимы как к статическим, так и к динамическим сценам.

Таким образом, формально процесс сегментации – это процесс выделения на изображении интересующих объектов. Выделение объектов может быть реализовано многими способами в зависимости от того, с какой целью это делается. Так, например, для распознавания объекта из ряда предложенных можно воспользоваться только контуром данного объекта, а можно провести сравнение по нескольким пунктам, для чего необходимо иметь полное изображение объекта. Цветовая сегментация обычно производится по распределению цвета искомого объекта в цветовом пространстве HSV.



## Порядок выполнения работы

1. Загрузить исходное изображение в соответствие с вариантом индивидуального задания.
2. Выполнить необходимые преобразования над изображением с целью его улучшения, включая использование методов и алгоритмов цветовой коррекции.
3. Выполнить операцию сегментации и представления в соответствии с вариантом индивидуального задания.
4. Результаты работы сохранять в директории: «.../DIP/Lab6/».

### Варианты индивидуального задания


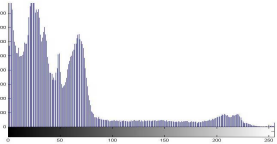

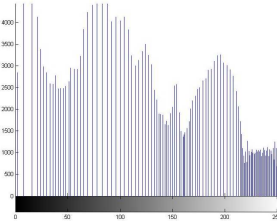
№	Файл изображения	Задание
1.	Lab6_Var1.png	Определить и выделить наибольший по площади синий объект на изображении.
2.	Lab6_Var2.png	Определить и выделить все квадратные оранжевые объекты и вывести их количество
3.	Lab6_Var3.png	Определить и выделить на изображении два красных объекта находящихся на максимальном расстоянии друг от друга
4.	Lab6_Var4.png	Определить и выделить на изображении наименее гладкий черный объект
5.	Lab6_Var5.png	Определить и выделить наибольший по периметру круглый желтый объект на изображении.
6.	Lab6_Var6.png	Определить и выделить на изображении зеленый объект с наибольшим углом наклона
7.	Lab6_Var7.png	Определить и выделить на изображении два зеленых объекта находящихся на минимальном расстоянии друг от друга
8.	Lab6_Var8.png	Определить и выделить на изображении наиболее гладкий красный объект
9.	Lab6_Var9.png	Определить и выделить на изображении розовый объект с наименьшим углом наклона
10.	Lab6_Var10.png	Определить и выделить на изображении бордовый объект с наибольшим углом наклона
11.	Lab6_Var11.png	Определить и выделить на изображении все прямоугольные черные объекты. Определить их количество.
12.	Lab6_Var12.png	Определить и выделить наименьший по площади фиолетовый объект на изображении

13.	Lab6_Var13.png	Определить и выделить наибольший по площади круглый оранжевый объект на изображении.
14.	Lab6_Var14.png	Определить и выделить все круглые белые объекты. Определить их количество.
15.	Lab6_Var15.png	Определить и выделить наименьший по площади серый объект на изображении

### **Вопросы для подготовки к защите**

1. Цветное изображение. Базовые операции и преобразования.
2. Цветовые модели.
3. Сегментация изображений.
4. Основные количественные параметры объектов на изображении.  
Площадь, периметр, расстояние, центр масс и т.д.

## Приложение. Пример оформления отчета по лабораторной работе

<p style="text-align: center; font-size: small;">МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ</p> <p style="text-align: center;"><b>«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»</b></p> <p style="text-align: center;">Кафедра оптико-электронных приборов и систем</p> <p style="text-align: center;">Отчет к лабораторной работе № _____ « _____ _____ » по курсу «Обработка оптических изображений»</p> <p style="text-align: center;">Выполнил: _____ Логин: _____ Группа: _____</p> <p style="text-align: center;">Проверил: _____</p> <p style="text-align: center;">Санкт-Петербург 2015</p>	<p style="text-align: center;"><b>Цель работы</b></p> <p style="text-align: center;"><b>Краткие теоретические сведения</b></p> <p style="text-align: center;"><b>Текст программы</b></p> <p style="text-align: center;"><b>Вывод</b></p>	<p style="text-align: center;"><b>Приложение. Результаты выполнения программы</b></p> <p>П.1 Исходное изображение и его гистограмма</p> <div style="display: flex; align-items: center;">   </div> <p>П.2 Изображение и его гистограмма после эквализации гистограммы</p> <div style="display: flex; align-items: center;">   </div>
--	---	--

Приложение отчета готовится **ТОЛЬКО ДЛЯ ЭЛЕКТРОННОЙ** версии. Для защиты **ПЕЧАТАТЬ ПРИЛОЖЕНИЕ** с результатами выполнения программы **НЕ НУЖНО**.

## Список литературы

1. Гонсалес Р., Вудс Р., Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
2. Краснящих А.В. Обработка оптических изображений. – СПб: НИУ ИТМО, 2012. – 129 с.
3. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Издание 3-е, исправленное и дополненное – М.: Техносфера, 2012. - 1104 с.
4. В.Т. Фисенко, Т.Ю. Фисенко, Компьютерная обработка и распознавание изображений: методические указания к лабораторным работам. - СПб: СПбГУ ИТМО, 2008. – 42 с.
5. Визильтер Ю. В., Желтов С. Ю., Князь В. А., Ходарев А. Н., Моржин А. В. Обработка и анализ цифровых изображений с примерами на LabVIEW IMAQ Vision. – М.: ДМК Пресс, 2007. – 464 с.
6. Цифровая обработка сигналов в среде matlab: лабораторный практикум в 2 ч. Ч. 2 / Т.В. Дашковская. – Новосибирск: СГГА, 2010. – 83 с.
7. Руднев А.А. Методы обработки видеоинформации в системах технического зрения промышленных роботов с применением Matlab/ А.А. Руднев, Д.А. Юдин – Белгород: Изд-во БГТУ, 2012. – 33 с.
8. Методы компьютерной обработки изображений // Под редакцией В.А. Сойфера. Москва: Физматлит, 2003
9. Сергеев В.В., Чичева М. А. Теория цифровой обработки сигналов и изображений. Самара: изд-во Самар. гос. аэрокосм. ун-та, Редсовет университета, 2013

## Содержание

1 Базовые функции для работы с цифровым изображением в среде MATLAB .....	4
Лабораторная работа №1: Основы цифрового представления оптических изображений.....	12
Лабораторная работа №2: Синтезирование изображений .....	15
Лабораторная работа №3: Основы методов пространственной обработки изображений.....	25
Лабораторная работа №4: Фильтрация изображений в частотной области ...	33
Лабораторная работа №5: Морфологическая обработка изображений.....	36
Лабораторная работа №6: Обработка и сегментация цветных изображений.	40
Приложение. Пример оформления отчета по лабораторной работе .....	44
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ .....	44
Список литературы .....	45

**Миссия университета** – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

---

## **КАФЕДРА ОПТИКО-ЭЛЕКТРОННЫХ ПРИБОРОВ И СИСТЕМ И ЕЕ НАУЧНО-ПЕДАГОГИЧЕСКАЯ ШКОЛА**

**Кафедра** создавалась в 1937-38 годах и существовала под следующими названиями:

- с 1938 по 1958 год - кафедра военных оптических приборов;
- с 1958 по 1967 год - кафедра специальных оптических приборов;
- с 1967 по 1992 год - кафедра оптико-электронных приборов;
- с 1992 года - кафедра оптико-электронных приборов и систем.

**Кафедру возглавляли:**

- с 1938 по 1942 год - профессор К.Е. Солодилов;
- с 1942 по 1945 год профессор А.Н. Захарьевский (по совместительству);
- с 1945 по 1946 год - профессор М.А. Резунов;
- с 1947 по 1972 год - профессор С.Т. Цуккерман;
- с 1972 по 1992 год - заслуженный деятель науки и техники РСФСР, профессор Л.Ф. Порфирьев;
- с 1992 по 2007 год - заслуженный деятель науки РФ, профессор Э.Д. Панков.

с 2007 года по настоящее время - почетный работник высшего профессионального образования, профессор В.В. Коротаев.

1938 по 1970 кафедра входила в состав оптического факультета.

В 1970 году кафедра вошла в состав факультета оптико электронного приборостроения, который в 1976 году был переименован в инженерно-физический факультет.

В 1998 г кафедра вошла в состав факультета оптико-информационных систем и технологий.

В 2015 году кафедра вошла в состав факультета лазерной и световой инженерии

Кафедра оптико-электронных приборов и систем (ОЭПиС) осуществляет подготовку профессионалов в области создания оптико-электронных и видеоинформационных приборов и систем, а также в области разработки их программного обеспечения.

Результаты научных исследований кафедры докладываются на ведущих мировых научных форумах, публикуются в виде научных статей и монографий.

Приборы, разработанные на кафедре, поставляются на предприятия России и на предприятия других стран.

Этот уникальный опыт передается нашим студентам.

На кафедре работают 6 докторов наук, профессоров, однако большую часть коллектива составляют молодые люди в возрасте от 18 до 35 лет, в том числе 11 молодых кандидатов наук.

Коллектив кафедры Оптико-электронных приборов и систем (ОЭПиС) – сформировавшаяся научная и научно-педагогическая школа, существующая с 1938 года.

За эти годы были подготовлены более тысячи специалистов, более ста докторов и кандидатов наук.

С 2007 г. заведующим кафедрой является почетный работник высшего профессионального образования Российской Федерации, профессор В.В. Коротаев.

Только в период с 2007 по 2015 год на кафедре были защищены 28 диссертаций на соискание ученой степени кандидата технических наук.

В 2012 году научно-педагогическая школа кафедры ОЭПиС «Оптико-электронное приборостроение» была внесена в реестр ведущих научных и научно-педагогических школ Санкт-Петербурга.

Подробная информация о кафедре ОЭПиС имеется на сайте кафедры:  
<http://oep.ifmo.ru/>

Васильев Александр Сергеевич  
Васильева Анна Владимировна

**Обработка оптических изображений.  
Лабораторный практикум**

**Учебно-методическое пособие**

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе