****

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» (МАИ)

**Кафедра** **806**  
**«Вычислительная математика и программирование»**

Отчёт по лабораторной работе №2   
  
**«Основы работы с цифровыми изображениями в Matlab»**

по курсу   
  
**«Цифровая обработка изображений»**

Выполнил: студент гр. М80-105СВ-25

Крючков А.В.

Принял:

Гаврилов К.Ю.

Москва  
2025

**Задание 7**

1. Исходное изображение ***Img2\_07\_1.jpg*** представляет собой черно-белую фотографию.

Создайте имитацию неравномерной засветки изображения, равномерно увеличивающейся снизу вверх.

Затем, используя функцию **imflatfield**, восстановите исходное изображение.

При восстановлении изображения подберите оптимальные значения параметров функции **imflatfield,** обеспечивающие наивысшую равномерность яркости по всей поверхности изображения.

При необходимости используйте функцию эквализации гистограммы.

1. Исходное изображение представлено в виде двумерной матрицы, записанной в формате Matlab в файле ***Z2\_07\_2.mat***.

Определите динамический диапазон яркостей изображения и подберите вид нелинейного преобразования, при котором можно будет различить все детали изображения как в центре, так и на краях.

Обеспечьте возможность изменения крутизны нелинейности преобразования, определяющей соотношения яркостей промежуточных тонов.

1. Использование для улучшения яркости и контраста изображений следующих функций:

**imajust** – реализует гамма-коррекцию,

**histeq** – реализует метод эквализации гистограммы,

**imlocalbrighten** – выполняет автоматическое выравнивания яркости

темных областей,

**imcontrast** – выполняет ручную подстройку яркости и контраста.

Дано изображение ***Img2\_07\_3.jpg***.

Постройте гистограмму яркости заданного изображения.

Используйте функцию **imajust** для коррекции контраста.

Постройте гистограмму яркости скорректированного изображения.

Сравните гистограммы яркостей обоих изображений и сделайте выводы.

Используйте функцию **histeq** для эквализации гистограммы.

Постройте гистограмму яркости после эквализации. Сравните гистограммы яркостей, полученные в результате использования функций **imajust**  и **histeq**.

Используйте функцию **imlocalbrighten** для улучшения контраста. Постройте гистограмму скорректированного изображения.

Используйте функцию **imcontrast** для улучшения яркости и контраста. Постройте гистограмму скорректированного изображения.

Задание 1

*%% Загрузка изображения*

inputImagePath = 'Img2\_07\_1.jpg';

originalGrayImage = im2double(rgb2gray(imread(inputImagePath)));

[height, width, *~*] = size(originalGrayImage);

*%% Генерация неравномерной засветки (градиент по вертикали)*

*% Создание матрицы засветки, увеличивающейся снизу вверх*

baseRowVector = ones(1, width);

shadingMatrix = zeros(height, width);

for currentRow = 1:height

  intensityFactor = (height - currentRow + 1)^2; *% Квадратичный градиент*

  shadingMatrix(currentRow, :) = intensityFactor \* baseRowVector;

end

*% Нормализация матрицы засветки*

shadingMatrix = shadingMatrix / max(shadingMatrix(:));

*% Применение засветки к исходному изображению*

shadedImage = originalGrayImage .\* shadingMatrix;

*% Нормализация затененного изображения*

shadedImage = shadedImage / max(shadedImage(:));

*%% Восстановление изображения с помощью коррекции неравномерной засветки*

*% Подбор параметра sigma для функции imflatfield*

sigma = 130;

correctedImage = imflatfield(shadedImage, sigma);

*%% Сохранение всех изображений*

resultsFolder = './results/';

imwrite(originalGrayImage, fullfile(resultsFolder, 'original\_image.png'));

imwrite(shadingMatrix, fullfile(resultsFolder, 'shading\_matrix.png'));

imwrite(shadedImage, fullfile(resultsFolder, 'shaded\_image.png'));

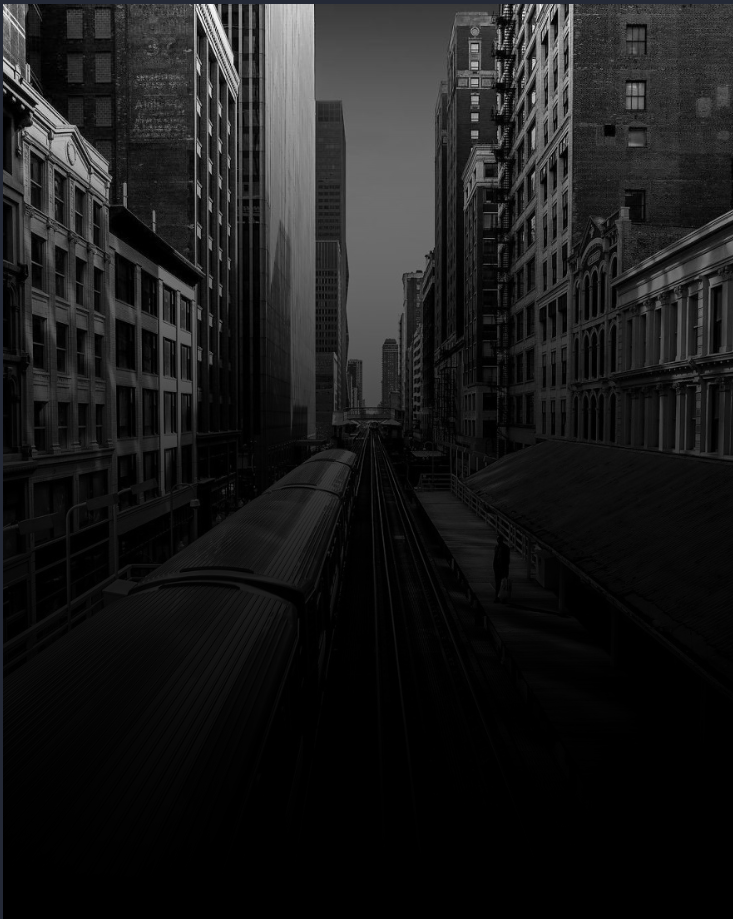
imwrite(correctedImage, fullfile(resultsFolder, 'corrected\_image.png'));

Результат:

original\_image.png



shaded\_image.png



corrected\_image.png



Задание 2

contrastParameter = 60;

resultsFolder = './results/';

imageData = load('Z2\_07\_2.mat', '-ASCII');

imwrite(imageData,  fullfile(resultsFolder, 'original\_image.png'));

*% Определение динамического диапазона яркостей исходного изображения*

imageMax = max(imageData(:));

imageMin = min(imageData(:));

fprintf('Динамический диапазон исходного изображения: от %f до %f\n', imageMin, imageMax);

*% Применение нелинейного (логарифмического) преобразования*

*% Формула: log(1 + contrastParameter \* I)*

*% Параметр contrastParameter позволяет управлять "крутизной" нелинейности*

logTransformed = log(1 + contrastParameter \* imageData);

*% Определение динамического диапазона преобразованного изображения*

logMin = min(logTransformed(:));

logMax = max(logTransformed(:));

fprintf('Динамический диапазон после лог-преобразования: от %f до %f\n', logMin, logMax);

*% Нормализация результата преобразования в диапазон [0, 1]*

logTransformed = logTransformed / logMax;

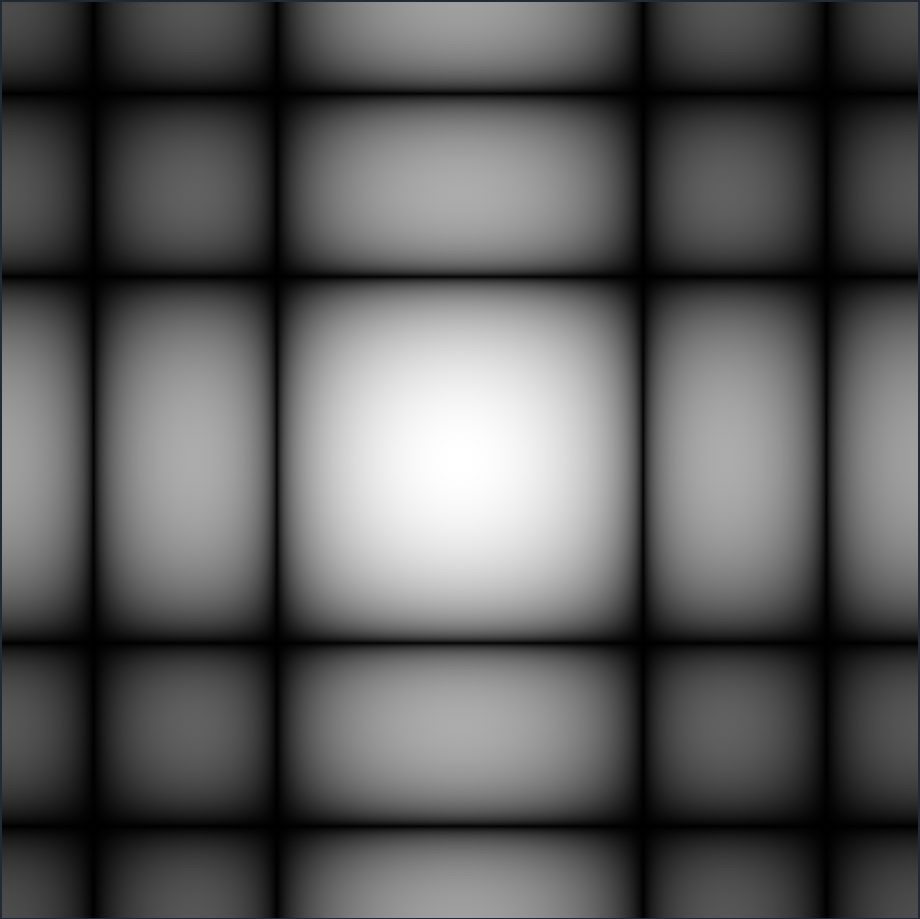
imwrite(logTransformed, fullfile(resultsFolder,'log\_transformed\_image.png'));

Результат:

original\_image.png



log\_transformed\_image.png



Задание 3

*%% Загрузка изображения*

imagePath = 'Img2\_07\_3.jpg';

originalImage = im2double(imread(imagePath));

figure();

imshow(originalImage, []);

title('Исходное изображение');

figure();

histogram(originalImage);

title('Гистограмма исходного изображения');

*%% Гамма-коррекция (imadjust)*

gammaValue = 0.3; *% Значение < 1 для осветления теней*

gammaRestoredImage = imadjust(originalImage, [], [], gammaValue);

figure();

imshow(gammaRestoredImage, []);

title(['Гамма-коррекция (gamma=', num2str(gammaValue), ')']);

figure();

histogram(gammaRestoredImage);

title('Гистограмма после гамма-коррекции');

*%% Эквализация гистограммы (histeq)*

histeqRestoredImage = histeq(originalImage);

figure();

imshow(histeqRestoredImage, []);

title('Эквализация гистограммы (histeq)');

figure();

histogram(histeqRestoredImage);

title('Гистограмма после эквализации');

*%% Автоматическое выравнивание яркости темных областей (imlocalbrighten)*

localBrightenRestoredImage = imlocalbrighten(originalImage);

figure();

imshow(localBrightenRestoredImage, []);

title('Автоматическое выравнивание яркости темных областей (imlocalbrighten)');

figure();

histogram(localBrightenRestoredImage);

title('Гистограмма после автоматического выравнивания яркости темных областей');

*%% Ручная настройка яркости/контраста (imcontrast)*

figure();

imshow(originalImage, []);

imcontrast;

input('Нажмите Enter после настройки в imcontrast...');

if ishandle(gca)

  processedImage = getimage;

  figure();

  imshow(processedImage, []);

  title('Изображение после ручной настройки (imcontrast)');

  figure();

  histogram(processedImage);

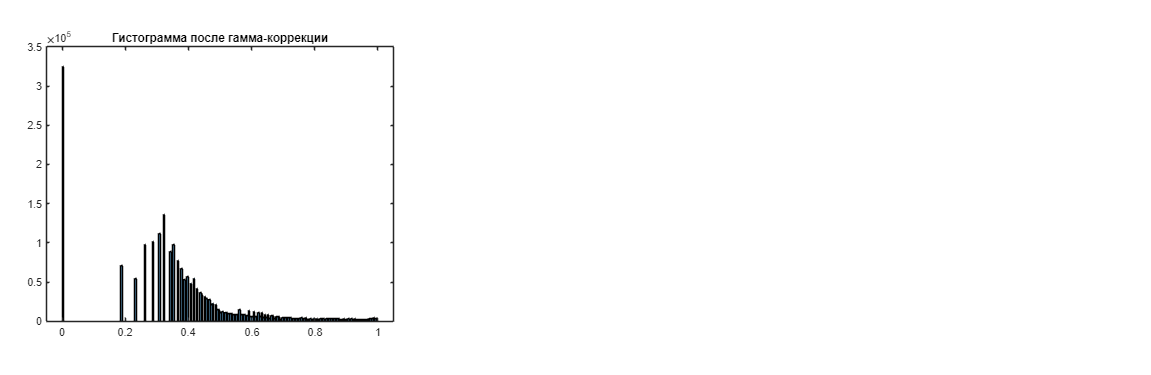
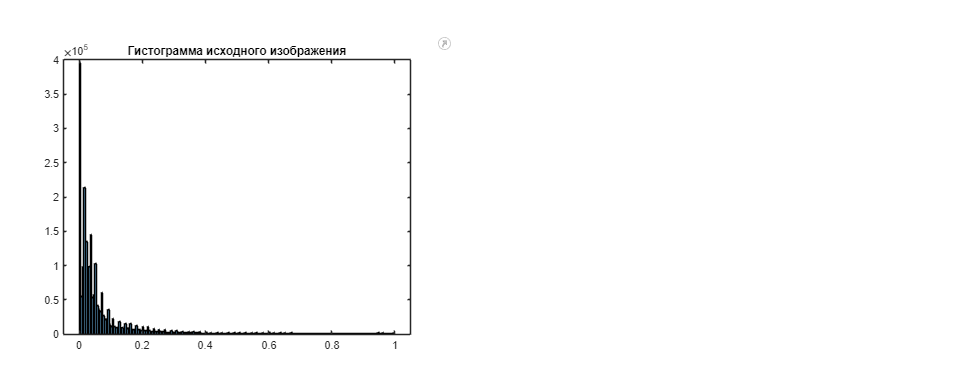
  title('Гистограмма после imcontrast');

else

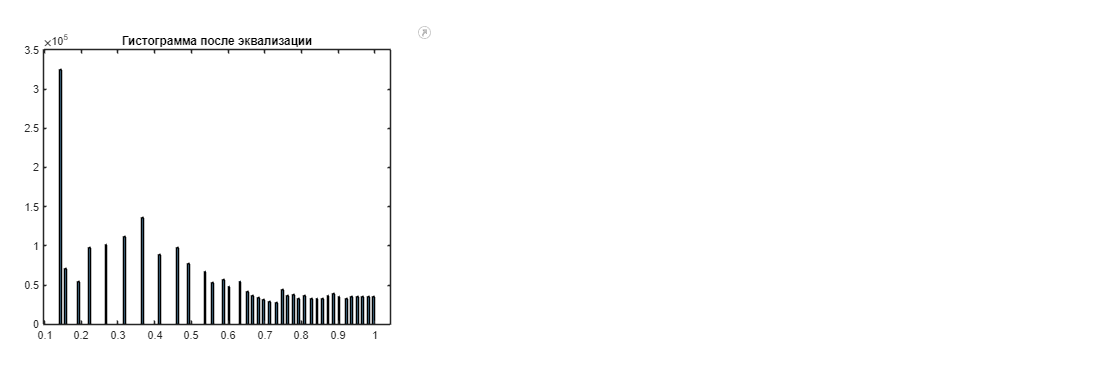
  error('Не удалось получить изображение из fig-окна');

end

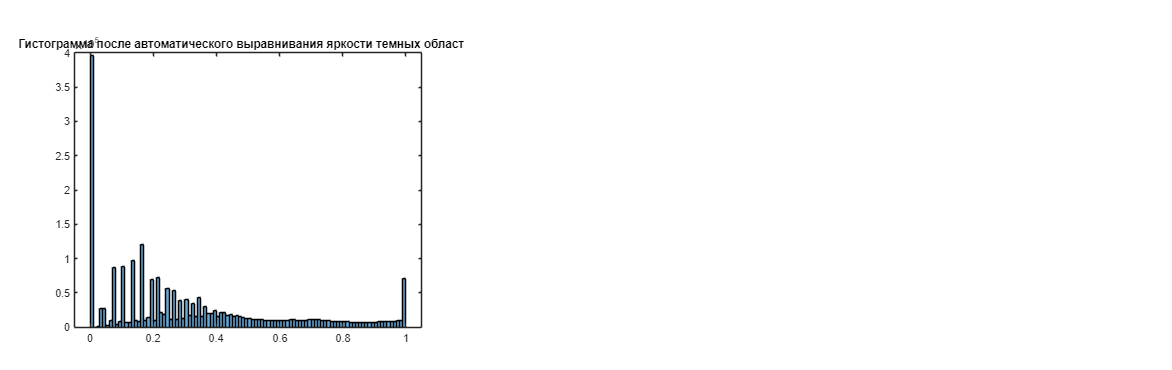
Результат:

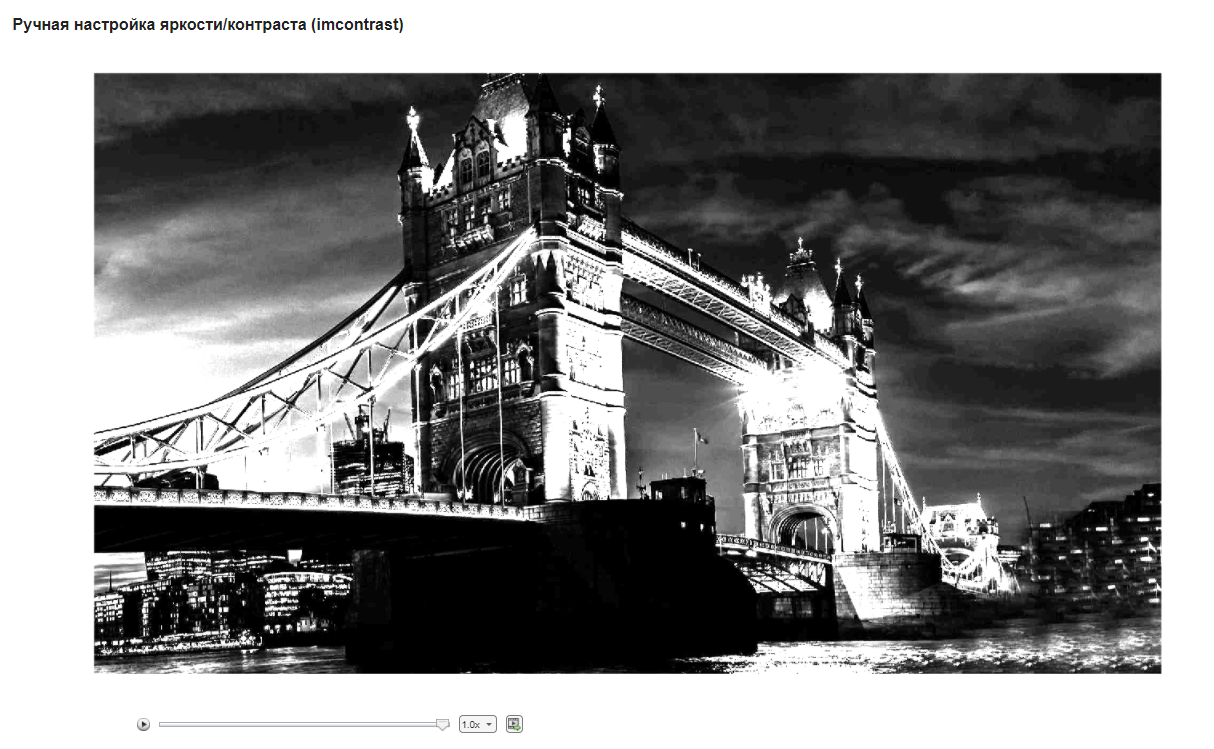


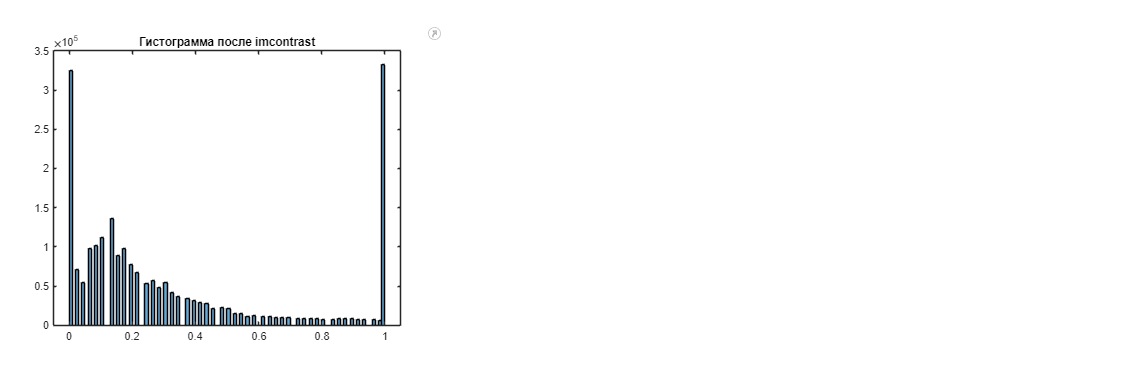












Вывод: Задание выполнено в полном объёме