ДОМАШНЯЯ РАБОТА №2
ПРЕОБРАЗОВАНИЕ ИЗОБРАЖЕНИЙ, НАЛОЖЕНИЕ ШУМА НА ИЗОБРАЖЕНИЕ И ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ

Цель задания

Получение практических навыков доступа к объектам графики в системе MatLab. Сравнительный анализ различных преобразований. Получение практических навыков использования двумерного преобразования Фурье при исследовании диапазона яркости изображения. Получение практических навыков наложения на изображение шума и фильтрации изображения.

Постановка задачи

1. Вывести изображение в графическое окно.
2. Осуществить прямое и обратное косинусное преобразование над изображением. Вывести в графическое окно результаты преобразований.
3. Осуществить прямое и обратное преобразование Фурье над изображением. Вывести в графическое окно результаты преобразований.
4. Определить коэффициент корреляции между исходным изображением и изображениями, полученными в результате обратных преобразований.
5. Выявить какое из полученных изображений менее всего отличается от оригинала.
6. Построить график зависимости спектра яркости от частоты.
7. На исходное изображение наложить различного рода шум, согласно варианту.
8. Произвести фильтрацию исходного изображения, согласно варианту.
9. Производить фильтрацию для трех различных масок фильтров.
10. С помощью коэффициента корреляции оценить действие каждого из фильтров с учетом размера маски. Построить графики зависимости коэффициента корреляции от размера маски.

Содержание отчета

1) Постановка задачи.
2) Код программы.
3) Выводы.

Методические указания

Форматы изображений

В пакете расширения Image Processing используются различные формы изображений и представляющих их файлов. Они даны в

приведенной ниже таблице. Соответствующие типы файлов могут быть прочитаны функцией imread.

формат	Глубина цвета	Особенности
BMP	1 -bit, 4-bit, 8-bit или 24-bit	Несжатые файлы, 4-bit и 8-bit – файлы с RLE-сжатием
CUR	1 -bit, 4-bit или 8-bit	Несжатые файлы
HDF	8-bit или 24-bit	8-bit растровые изображения с цветовой картой и без цветовой карты; 24-bit растровые изображения
ICO	1 -bit, 4-bit или 8-bit	Несжатые файлы
JPEG	8 или 24-bit	Сжатые по стандарту JPEG с разной степенью компрессии файлы
PCX	1 -bit, 8-bit или 24-bit	Формат известной программы Paint Brush
PNG	1-bit, 2-bit, 4-bit, 8-bit, 16-bit, 24-bit или 48-bit	1-bit, 2-bit, 4-bit, 8-bit, и 16-bit полутоновые изображения; 8-bit и 16-bit палитровые изображения; 24-bit и 48-bit полноцветные изображения
TIFF	1 -bit, 8-bit, 16-bit, 24-bit или 48-bit	1 -bit, 8-bit, и 24-bit несжатые файлы; 1 -bit, 8-bit, 16-bit, и 24-bit с packbits-сжатием; 1 -bit с CCITT (МККТТ)-сжатием; 16-bit полутоновые; 16-bit палитровые; 48-bit полноцветные изображения
XWD	1 -bit или 8-bit	Графика X-Windows

Получение изображения из графического объекта – getimage

Функция A = getimage(h) возвращает изображение A, содержащееся в графическом объекте с идентификатором h, причем исходный графический объект может быть графиком функции, изображением или текстурной поверхностью с идентификатором h.

[X, y, A] = getimage(h) – возвращает дополнительно два двухэлементных вектора XData и YData, содержащих диапазоны изменения координат по осям X и У.

[…] = getimage – возвращает дополнительную информацию о текущем графическом объекте и эквивалентна функции [...] = getimage (gca).

[..., A, flag] = getimage(h) – возвращает дополнительно целое число - флаг, – определяющее тип исходного изображения h. Ниже приведены допустимые значения для переменной flag:

- f1ag = 0 – отсутствие исходного изображения, A – пустая матрица.
- flag = 1 – полутоновое исходное изображение со значениями яркости в стандартных диапазонах ([0, 1] для

формата представления данных double, [0, 255] для uint8, [0,65535] для uint16);

- f1ag = 2 – палитровое изображение;
- flag = 3 – полутоновое изображение с нестандартным диапазоном яркости;
- flag = 4 – полноцветное изображение (RGB).

Вывод изображения на экран – imshow

Функция imshow служит для *вывода изображения* на экран и используется в ряде форматов:

- imshow(I, n) – выводит на экран полутоновое изображение I с использованием n дискретных уровней серого. Если параметр n опущен, то функция imshow по умолчанию использует 256 градаций серого 24-битового режима или 64 градации серого для других режимов;
- imshow(I, [low high]) – выводит на экран полутоновое изображение I с повышенной контрастностью. Все элементы входного массива, имеющие значение low или меньше, отображаются черным цветом, все элементы со значениями больше или равными high отображаются белым цветом. Значения между low и high отображаются промежуточными оттенками серого. Если в качестве входного аргумента задана пустая матрица ([]) для [low high], функция imshow для low и high использует следующие значения по умолчанию: [min(I(:)) max(I(:))];
- imshow(BW) – выводит на экран бинарное изображение BW. Нулевые элементы входного массива отображаются черным цветом, 1 – белым;
- imshow(X, map) – выводит на экран палитровое изображение X с цветовой картой tap;
- imshow(RGB) – выводит на экран полноцветное изображение RGB.
- imshow(display_option) – выводит на экран изображение, используя параметры 'truesize' или 'notruesize' в позиции display option.
- imshow(x, y ,A) – дополнительно использует двухэлементные векторы x и y для установки пространственной системы координат с использованием параметров XData и YData;
- imshow filename – выводит на экран изображение из файла с именем filename. При этом автоматически вызывается функция imread для считывания изображения из внешнего

файла;

- h = imshow(...) – возвращает дескриптор данного изображения.

Вывод на экран нескольких изображений в одном окне – subimage

Функция subimage может быть использована для вывода на экран нескольких изображений в одном окне, даже если эти изображения имеют разные цветовые карты.

- subimage(X, map) – выводит на экран палитровое изображение X с цветовой картой tap в текущем масштабе.
- subimage(I) – выводит на экран полутоновое изображение I в текущем масштабе.
- subimage(BW) – выводит на экран бинарное изображение BW.
- subimage(RGB) – выводит на экран полноцветное изображение RGB.
- subimage(x,y...) – выводит на экран изображение с использованием заданной системы координат.
- h = subimage(...) – возвращает дескриптор графического объекта.

Пример: Вывод в одном графическом окне двух изображений

```
[X1, map1] = imread('m83.tif');
[X2, map2] = imread('trees.tif');
subplot(1, 2, 1),
subimage(X1, map1);
subplot(1, 2, 2),
subimage(X2, map2)
```

Чтение изображения из файла – imread

Функция A = imread(filename.fmt)

читает из файла с именем filename полутоновое или полноцветное изображение и создает A. Если исходное изображение полутоновое, то A - двумерный массив, если исходное изображение полноцветное, то A – рехмерный массив размера m x n x 3.

Другие формы этой функции:

- [X, map] = Imread(filename, fmt) – читает из файла с именем filename палитровое изображение в массив A с цветовой картой map;
- [...] = imread(filename) – пытается определить

информацию о формате файла по его содержанию. Параметры filname и fmt были подробно рассмотрены в описании функции imfinfo;

- [...] = imread(... ,idx) – читает одно изображение из TIFF файла. idx – целое число – номер изображения по порядку.

Описываемая функция имеет ряд особенностей для PNG-файлов, содержащих *прозрачные пикселы* (хотя и не всегда). Прозрачные пикселы, если они существуют, идентифицируются одним или двумя компонентами: часть данных прозрачности и альфа-канал. Часть данных прозрачности определяет прозрачные пикселы напрямую, например, если часть данных прозрачности 8-битового изображения равна 0.5020, то все пикселы изображения с цветом 0.5020 будут выведены на экран как прозрачные. Заметим, что PNG-файл может содержать вместо двух только один компонент – альфа-канал.

Альфа-канал представляет собой массив с таким же числом пикселов, как и исходное изображение, который определяет признак прозрачности каждого пиксела (прозрачный или непрозрачный). И наконец, последний компонент PNG-файла – это данные цвета фона, которые определяют значение цвета, «просвечивающегося» из-под прозрачных пикселов. Ниже описывается поведение IPT по умолчанию при чтении PNG-изображений, содержащих или часть данных прозрачности, или альфа-канал.

- [...] = imread(..., 'BackgroundColor' ,bg) – считывает изображение из PNG-файла и комбинирует пикселы прозрачности против определенного цвета. Форма параметра bg зависит от формата входного файла. Если входное изображение палитровое, то параметр bg должен быть целым числом порядка [1, P], где P – длина массива цветовой карты. Если входное изображение полутоновое, то параметр bg должен быть целым числом порядка [0,1]. Если входное изображение полноцветное, то параметр bg должен быть трехэлементным вектором со значениями порядка [0,1].
- [A, map, alpha] = imread(...) – возвращает A – шаблон для указания способа, который используется для определения информации о прозрачности.
- [A, map, alpha] = imread(filename) или [A, map, alpha] = imread(filename, fmt) считывает изображение из PNG-файла, если не применяется комбинирование и альфа-канал сохраняется отдельно от изображения
- [...] = imread(……, ref) – считывает одно изображение из HBP-файла. Параметр ref – целое число, определяющее справочное число идентифицирующее изображение.

- [...] = imread(….., idx) – считывает одно изображение из CUR- и ICO-файлов. Параметр idx – это целое число, определяющее порядок изображения в файле.

Запись изображения в файл — imwrite

Для записи массива с изображением в файл служит функция imwrite. Она имеет следующие формы:

- imwrite(A, filename, fmt) – записывает изображение в файл с именем filename в формате fmt из массива A. A может быть матрицей размера M x N для полутонового изображения и массивом размера M x N x 3 для полноцветного изображения. Если A относится к классу uint8 или uint16, то функция imwrite записывает фактические значения из массива в файл. Если A относится к классу double, то функция imwrite перемасштабирует значения в исходном массиве перед записью по формуле uint8(round(255*A)). При этом числа с плавающей запятой в диапазоне [0,1] преобразуются в 8-битовые целые числа в диапазоне [0,255].
- Imwrite(X, map, filename, fmt) – записывает палитровое изображение в файл с именем filename в формате fmt из массива X и соответствующей цветовой карты tap. Если X относится к классу uint8 или uintl16 то функция imwrite записывает фактические значения из массива в файл. Если X относится к классу double, то функция imwrite смешает значения в исходном массиве перед записью по формуле uint8(X-l). Массив tap должен быть цветовой картой MATLAB класса double функция imwrite перемасштабирует исходные значения массива пар по формуле uint8(round(255*map)). Заметим, что большинство графических файлов не поддерживают цветовых карт с количество ячеек больше, чем 256.
- Imwrite(fi1ename) – аналогична описанным выше функциям, а формат файла определяется по расширению filename.

В таблице ниже приведены типы изображений, которые могут быть записаны функцией imwrite:

Формат	Тип изображения
BMP	8-bit палитровые несжатые и 24-bit несжатые файлы
HDF	8-bit растровые изображения с цветовой картой и без цветовой карты 24-bit растровые изображения, несжатые или с RLE-или JPEG-сжатием
JPEG	8- или 24-bit изображения, причем палитровые изображения

	конвертируются в полноцветные
PCX	8-bit изображения
PNG	1 -bit, 2-bit, 4-bit, 8-bit, 1 6-bit полутоновые, 8-bit, и 16-bit полутоновые с альфа-каналом, 1 -bit, 2-bit, 4-bit, 8-bit палитровые, 24-bit и 48-bit с альфа-каналом и без него
TIFF	1-bit, 8-bit, 16-bit, 24-bit и 48-bit
XWD	8-bit

- imwrite(.... Paraml, Va11, Param2, Va12...) – определяет параметр, который контролирует различные характеристики выходного файла. Установка параметра может быть сделана для файлов HDF, PNG, JPEG и TIFF. Параметры filename и fmt являются массивами символов. Ниже приведены допустимые значения параметра fmt:

Значение параметра fmt	Название формата
‘bmp’	Windows Bitmap (BMP)
‘hdf’	Hierarchical Data Format (HDF)
‘jpg' или 'jpeg'	Joint Photographic Experts Group (JPEG)
'png'	Portable Network Graphics (PNG)
‘tif’ или 'tiff’	Tagged Image File Format (TIFF)
‘pcx'	Windows Paintbrush (PCX)
‘xwd'	X Windows Dump (XWD)

Двумерное дискретное косинусное преобразование – dct2

Функция B = dct2(A) возвращает результат *двумерного дискретного косинусного преобразования* для матрицы A. Матрица B имеет тот же размер, что и матрица A, и представляет коэффициенты дискретного косинусного преобразования.
Функция B = dct2(A, m, n) или B = dct2(A,[m n]) обеспечивает двумерное косинусное преобразование матрицы A размера m x n после преобразования. Если размер матрицы A меньше этого размера, она дополняется нулевыми элементами до заданного размера.

Двумерное обратное дискретное косинусное преобразование – idct2

Функция B = idct2(A) осуществляет двумерное обратное дискретное косинусное преобразование для матрицы A и возвращает его результат в виде матрицы B.

Функции B = idct2(A, m, n) или B = idct2(A, [m n]) осуществляют двумерное обратное дискретное косинусное преобразование с размером матриц A и B m x n, дополняя при этом матрицу A нулевыми элементами, если ее размер меньше заданного.
Для многих A = idct2(dct2(A)) эквивалентно A с погрешностью до округления. Матрица A может содержать элементы класса double или любого класса integer. Матрица B имеет элементы класса double.

MATLAB-функции быстрого преобразования Фурье

В состав функций пакета Image Processing Toolbox входит несколько MATLAB-функций, которые мы уже рассматривали, в частности при описании пакета Signal Processing Toolbox. Отметим эти функции:

- fft2 – двумерное быстрое преобразование Фурье;
- fftn – n-мерное быстрое преобразование Фурье;
- fftshift – перегруппировка выходного массива при быстром преобразовании Фурье;
- ifft2 – двумерное обратное быстрое преобразование Фурье;
- ifftn – n-мерное обратное быстрое преобразование Фурье.

Наложение на изображение шума – imnoise

Для отработки методов удаления шума с изображений нужны тестовые изображения с шумовыми компонентами. Встречается и применение таких изображений в художественных целях, например для представления изображений, полученных издалека после прохождения радиотехнических трактов.
Функция J = imnoise(I, type) добавляет к изображению I сигнал с указанием класса шума в виде строки:

- 'gaussian' – гауссовый белый шум;
- 'salt & pepper' – шум в виде включенных или выключенных пикселов;
- 'speckle' – мультипликативный шум.

J = imnoise(I, type, parameters) позволяет задать дополнительно параметры шума.
J = imnoise(I, 'gaussian', m, v) добавляет гауссовый белый шум со средним значением m и отклонением v (по умолчанию m=0 и v= 0.01). Этот вид шума виден как на светлых, так и на темных областях изображения в виде характерных точек – «сыпи».
J = imnoise(I, 'salt & pepper' , d) позволяет задать плотность шума «соль и перец» d (по умолчанию 0.05) для включенных и выключенных пикселов. Этот вид шума также может вызывать

появление шумовой «сыпи» как на светлых, так и темных участках изображения.
J = imnoise(I, 'speckle', v) добавляет мультипликативную компоненту шума, так что J = I + n*I, где n – равномерно распределенный шум со средним значением 0 и среднеквадратичным отклонением v (по умолчанию 0.04). Этот шум не виден на темных участках изображения, но проявляется на его светлых участках.
Входное изображение должно быть представлено матрицей классов uint8, uint16 или double. Выходной сигнал имеет тот же класс, что и входной.
В приведенном ниже примере показано исходное изображение и три зашумленных изображения:

```
I = imread('saturn.tif’); figure; subplot(2, 2, l); imshow(I);
I1 = imnoise(I, 'salt & pepper', 0.02); subplot(2, 2, 2); imshow (I1);
I2 = imnoise(I, 'gaussian', 0.2, 0.03); subplot(2, 2, 3); imshow(I2);
I3 = imnoise(I,  'speckle', 0.03); subplot(2, 2, 4); imshow (13);
```

Исходное изображение может быть не только полутоновым, но и цветным.

ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЯ

Медианная фильтрация изображения — medfilt2

Эффективным средством фильтрации шума типа «соль и перец» – «salt and pepper» (см. выше) является медианная фильтрация. Для данного класса шума эта нелинейная фильтрация обычно дает лучшие результаты, чем фильтрация на основе операции свертки. Она является частным случаем ранговой фильтрации, описанной ниже.
Функция B = medfilt2(A, [m n]) фильтрует матрицу исходного изображения, используя маску фильтра размера m x п. Центральный пиксел маски получают усреднением всех ее пикселов. Маска применяется нерекурсивно ко всему изображению. Если изображение не удовлетворяет условиям применения маски, оно дополняется нулевыми элементами при классе A uint8 или единицами при классе A – double.
B = medfilt2 (A) задает медианную фильтрацию при маске, содержащей 3x3 пиксела.
B = medfiIt2(A,' indexed',...)– осуществляет медианную фильтрацию для палитрового изображения.
Входное изображение представляется матрицей A, принадлежащей к классам uint8, uint16 или double (если используется индексированное

изображение, то A не может быть класса uint16). Класс выходной матрицы B – тот же, что класс входной матрицы A.
Рассмотрим построение исходного изображения, наложение на него сильной шумовой компоненты и осуществление медианной фильтрации для маски 3 x 3 и 8 x 8 пикселов:

I = imread('saturn.tif'); figure; subp1ot(2, 2, l); imshow(I);
I1 = imnoise(I, 'salt & pepper', 0.05); subplot(2, 2, 2); imshow (I1);
I2 = medfilt2(Il, [8 8]); subplot(2, 2, 3); imshow(I2);
I3 = medfilt2(Il); subplot(2, 2, 4); imshow (13);
В данном примере при маске 3x3 пиксела можно (правда, с трудом) усмотреть небольшое смазывание изображения. Однако при маске в 8x8 пикселов отфильтрованное изображение практически неотличимо от оригинала.

Ранговая фильтрация – ordfilt2

Для осуществления *ранговой фильтрации* полутоновых изображений служит функция
B = ordfilt2 (A, order, domain).

Пикселы исходного изображения, соответствующие ненулевым элементам маски фильтра domain, сортируются в порядке возрастания. Пикселу изображения A, соответствующему центральному элементу маски, присваивается значение с порядковым номером order. Это делается нерекурсивно. Чтобы размеры A и B были одинаковы, матрица A временно может дополняться элементами по правилу, описанному для функции medfi1t2.
Функция B = ordfilt2(A, order, domain, S) работает аналогично, но позволяет задать матрицу S, из которой берутся недостающие для реализации метода ненулевые элементы.
B = ordfilt2(..., padopt) позволяет задать дополнительный граничный параметр padopt: 'zeros' (по умолчанию) или 'symmetric'.
Матрица A может быть классов uint8, uint16 или double. Матрица B имеет тот же класс, что и матрица A, за исключением использования аддитивной формы функции (с матрицей S), когда B – double.
Самым интересным свойством этого алгоритма является возможность создания эффектов расфокусировки *(эрозии)* и фокусировки (уточнение, наращивания фона и т. д.) изображения. В первом случае центральный пиксел маски надо выбрать как пиксел минимальный, а во втором случае – как пиксел максимальный (по

порядку в маске). Изменяя положение центрального пиксела маски, свойства ранговой фильтрации можно менять в широких пределах.
Эти эффекты для изображения монет наглядно иллюстрирует следующий пример:

```
I = imread('eight.tif); figure; subplot(1, 3, 1); imshow(I);
I2= ordfilt2 (I, l, ones(4, 4)); subplot(1, 3, 2);  imshow(I2);
I3= ordfilt2 (I, 16, ones(4, 4)); subplot(1, 3, 3);  imshow (I3);
```

А теперь рассмотрим пример работы ранговой фильтрации с изображением, искаженным шумом в виде мелких точек (пикселов), светлых на темном фоне и темных на светлом. Зададим в одном случае эрозию изображения, а в другом фильтрацию по центральному пикселу в маске (8-му):

```
I = imread('saturn.tif'); figure: subp1ot(2,2,l); imshow(I);
I1=imnoise(I, 'salt & pepper', 0.05): subplot(2,2,2); imshow (I1);
I2= ordfilt2 (I1, l, ones(4, 4)); subplot(2,2,3);  imshow(I2);
I3= ordfilt2 (I1, 8, ones(4, 4)); subplot(2,2,4);  imshow (I3);
```

Адаптивная фильтрация Винера – wiener2

функция J = wiener2(I, [m n], noise) реализует *адаптивную филытрацию Винера* для изображения, заданного матрицей A. Вектор [m n] задает размеры скользящего окна фильтра. При осуществлении фильтрации учитываются статистические особенности изображения в пределах окна, вчастности среднее значение яркости и ее среднеквадратическое отклонение. Параметр noise задает мощность гауссовского белого шума. Он может быть опущен, и тогда мощность шума будет задаваться автоматически.
Функция [J, noise] = wiener2(I, [m n]) дополнительно возвращает оценку мощности гауссовского шума.
Данный метод фильтрации обычно применяется для адаптивной фильтрации изображений с подавлением гауссовского белого шума. Матрица исходного изображения I и матрица выходного изображения J относятся к классам uint8, uintl6 или double.
Следующий пример демонстрирует степень эффективности адаптивной фильтрации применительно к очистке от шума изображения:

```
I = imread('saturn.tif'); figure; subplot(1, 3, 1); imshow(I);
J = imnois(I, 'gaussian' ,0 ,0.01); subplot(1, 3, 2);  imshow(J);
K = wiener2(J, [8 8]); subplot(1, 3, 3); imshow (K);
```

Варианты

№	Вид шума, тип фильтра	Вид шума, тип фильтра	Вид шума, тип фильтра
1	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, ранговая фильтрация	Мультипликативный шум, фильтрация Винера
2	Мультипликативный шум, медианная фильтрация	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера
3	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация	Гауссовый шум, фильтрация Винера
4	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, медианная фильтрация
5	Гауссовый шум, фильтрация Винера	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация
6	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, ранговая фильтрация
7	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, ранговая фильтрация	Мультипликативный шум, фильтрация Винера
8	Мультипликативный шум, медианная фильтрация	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера
9	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация	Гауссовый шум, фильтрация Винера
10	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, медианная фильтрация
11	Гауссовый шум, фильтрация Винера	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация
12	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, ранговая фильтрация
13	Гауссовый шум,	Шум в виде	Мультипликативный

	медианная фильтрация	включенных пикселей, ранговая фильтрация	шум, фильтрация Винера
14	Мультипликативный шум, медианная фильтрация	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера
15	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация	Гауссовый шум, фильтрация Винера
16	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, медианная фильтрация
17	Гауссовый шум, фильтрация Винера	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация
18	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, ранговая фильтрация
19	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, ранговая фильтрация	Мультипликативный шум, фильтрация Винера
20	Мультипликативный шум, медианная фильтрация	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера
21	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация	Гауссовый шум, фильтрация Винера
22	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, медианная фильтрация
23	Гауссовый шум, фильтрация Винера	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация
24	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, ранговая фильтрация
25	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, ранговая фильтрация	Мультипликативный шум, фильтрация Винера
26	Мультипликативный шум, медианная фильтрация	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера
27	Шум в виде	Мультипликативный	Гауссовый шум,

	включенных пикселей, медианная фильтрация	шум, ранговая фильтрация	фильтрация Винера
28	Гауссовый шум, ранговая фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, медианная фильтрация
29	Гауссовый шум, фильтрация Винера	Шум в виде включенных пикселей, медианная фильтрация	Мультипликативный шум, ранговая фильтрация
30	Гауссовый шум, медианная фильтрация	Шум в виде включенных пикселей, фильтрация Винера	Мультипликативный шум, ранговая фильтрация