

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 31.12.2020 13:36:44
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d895f1d11abbf73e943df6a4851fd156d089

МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии



УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

2017 г.

ОСНОВЫ МЕДИКО-БИОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ

Методические указания к практическим занятиям по дисциплине
«Организация медико-биологических исследований»

Курск 2017

УДК 004.93:61

Составители: О.В. Шаталова, К.Д.А. Кассим.

Рецензент

Доктор технических наук, профессор А.Ф. Рыбочкин

Основы медико-биологических исследований:
методические указания к практическим занятиям / Юго-Зап. гос. ун-
т; сост.: О.В. Шаталова, К.Д.А. Кассим. Курск, 2017. 28 с.

Предназначено для студентов направления подготовки 12.04.04 «Биотехнические системы и технологии» по дисциплине «Организация медико-биологических исследований». Может быть использована аспирантами, обучающимися по направлениям 05.11.13 – Системный анализ, управление и обработка информации и 05.11.17 – Приборы, системы и изделия медицинского назначения.

Текст печатается в авторской редакции

Подписано в печать 3.05.17. Формат 60×84 1/16. Бумага офсетная.

Усл. печ. л. 1,63. Уч.-изд. л. 1,5. Тираж 100 экз. Заказ 896.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

Практическая работа №1 Обработка двумерных изображений в MATLAB

Обработка изображений в MATLAB

Рассмотрим некоторые основы обработки изображений. В качестве примера покажем статистическую обработку изображения, а также предварительную обработку для получения улучшенного результата. Предобработка включает формирование равномерного фона на изображении и преобразование изображения в бинарное. Таким образом, обработка изображения состоит из некоторых основных шагов:

Шаг 1: Считывание и отображение изображения.

Шаг 2: Оценка и аппроксимация значений пикселей фона.

Шаг 3: Аппроксимация и просмотр поверхности фона.

Шаг 4: Создание изображения с равномерным фоном.

Шаг 5: Повышение контраста изображения.

Шаг 6: Создание бинарного изображения.

Шаг 7: Определение числа объектов на изображении.

Шаг 8: Анализ изображения.

Шаг 9: Отображение матрицы меток как псевдоцветового индексного изображения

Шаг 10: Измерение свойств объектов на изображении.

Шаг 11: Вычисление статистических данных об объектах на изображении.

Шаг 1: Считывание и отображение изображения

Считаем и визуализируем полутоновое изображение «rice.png» (рисунок 1.1).

```
I=imread('rice.png');  
imshow(I)
```

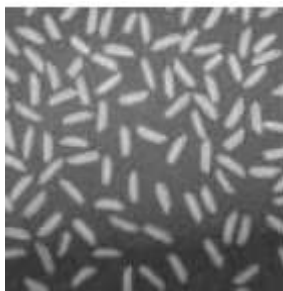


Рисунок 1.1 – Полутоновое изображение риса «rice.png»

Шаг 2: Оценка и аппроксимация значений пикселей фона

Рассматриваемое изображение характеризуется неравномерной засветкой фона по полю изображения. Поэтому сначала с помощью операции морфологического раскрытия оценим интенсивность фона. При выполнении операции морфологического раскрытия используется структурный элемент.

В примере используется функция «`imopen`», которая выполняет морфологическое раскрытие. При этом также используется функция «`strel`» для создания структурного элемента в виде диска с радиусом 15.

```
background = imopen(I, strel('disk', 15));
```

Для просмотра и оценки фона используются следующие команды

```
figure, imshow(background)
```

Шаг 3: Аппроксимация и просмотр поверхности фона

Используем команду «`surf`» для отображения поверхности фона. С помощью команды «`surf`» создается цветная параметрическая поверхность, которая дает возможность просматривать прямоугольную область исследуемого изображения. Функция «`surf`» работает с данными, которые представлены в формате «`double`». Поэтому, прежде чем применять эту функцию, данные необходимо перевести в формат «`double`».

```
figure,  
surf(double(background(1:8:end, 1:8:end))),
```

```
zlim([0 255]);
set(gca, 'ydir', 'reverse');
```

В данном примере отображается только каждый восьмой пиксель по каждому направлению. Существует также возможность установки масштаба и других параметров.

На основе этого можно проводить анализ фона исследуемого изображения (рисунок 1.2).

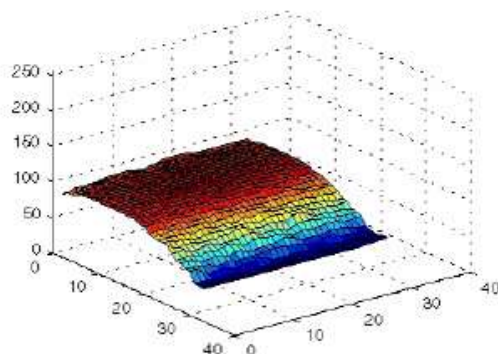


Рисунок 1.2 – Отображения поверхности фона

Шаг 4: Создание изображения с равномерным фоном

Для создания изображения с более равномерным фоном, вычтем изображение фона «background» из исходного изображения I.

```
I2=imsubtract(I,background);
```

Отобразим полученное изображение с более равномерным фоном.

```
figure, imshow(I2)
```

Шаг 5: Улучшение контраста на обрабатываемом изображении

После вычитания полученное изображение будет иметь более равномерный фон и будет более темным. Используем функцию «imadjust» для повышения контраста изображения.

```
I3=imadjust(I2);
```

Функция «`imadjust`» увеличивает контраст изображения путем растяжения значений интенсивностей динамического диапазона. Для более детальной информации см. описание функции «`imadjust`» системе Matlab. Отобразим улучшенное изображение I3.

```
figure, imshow(I3);
```

Шаг 6: Создание бинарного изображения

Бинарное изображение можно создать, используя функцию «`thresholding`» (рисунок 1.3). Однако функция «`graythresh`» автоматически определяет подходящий порог, который используется для преобразования полутонового изображения в бинарное. Функция «`im2bw`» выполняет это преобразование.

```
level=graythresh(I3);  
bw=im2bw(I3,level);  
figure, imshow(bw)
```



Рисунок 1.3 – Бинарное изображение

Возвращаемое функцией «`im2bw`» бинарное изображение «`bw`» представлено в формате «`logical`». В этом можно убедиться, воспользовавшись функцией «`whos`». Приложение «`ImageProcessingToolbox`» использует логические массивы для представления бинарных изображений.

```
whos
```

Реакцией системы MATLAB на эту функцию представлена в таблице 1.1

Таблица 1.1 – Реакция системы MATLAB на функцию «whos»

Name	Size	Bytes	Class
I	256x256	65536	uint8 array2
I2	256x256	65536	uint8 array
I3	256x256	65536	uint8 array
background	256x256	65536	uint8 array
bw	256x256	65536	logical array
level	1x1	8	doublearray

В общей сложности используется 327681 элемента, которые занимают 327688 байт.

Шаг 7: Определение числа объектов на изображении

После преобразования изображения в бинарное, можно использовать функцию «bwlabel» для определения числа объектов (зерен риса) на изображении. Функция «bwlabel» отмечает все компоненты на бинарном изображении «bw» и возвращает их число в виде значения «numObjects».

```
[labeled, numObjects]=bwlabel (bw, 4) ;
numObjects
ans=101
```

Точность результата зависит от некоторых факторов, включая

1. размер объектов;
2. соприкасаются ли между собой объекты (в этом случае они могут определяться как один объект);
3. точность аппроксимации фона.
4. выбор связности.

Шаг 8: Анализ матрицы меток

Для проведения анализа формирования матрицы меток, возвращаемой функцией «bwlabel», рассмотрим значения пикселей изображения. Есть несколько путей получения значений пикселей изображения. Например, используя функцию «imshow», можно просматривать небольшие порции изображения. Другой путь

состоит в использовании приложения «PixelRegiontool» для просмотра значений пикселей. Отобразим матрицу меток с использованием функции «imshow»,

```
figure, imshow(labeled);
```

Запуск приложения «PixelRegiontool»/

```
impixelregion
```

По умолчанию оно автоматически связывает себя с изображением в текущем окне просмотра. Приложение «PixelRegiontool» рисует прямоугольник с центром в видимой части изображения. Расположение этого прямоугольника определяет какие пиксели будут отображаться в «PixelRegiontool». При перемещении прямоугольника значения отображаемых в окне «PixelRegiontool» пикселей обновятся.

На следующем изображении представлен «ImageViewer» с прямоугольником «PixelRegion», размещенным на границе двух рисовых гранул (рисунок 1.4). Отметим, что значения пикселей фона и отдельных гранул являются одинаковыми. Так, в частности, значение фона равно 0.

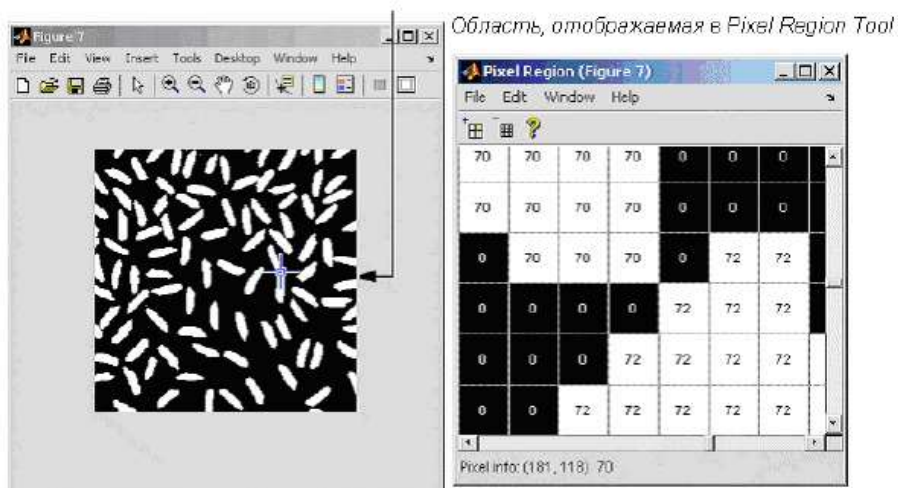


Рисунок 1.4 – Просмотр матрицы средствами «PixelRegiontool»

Шаг 9: Отображение матрицы меток в виде псевдоцветного индексного изображения

Один из возможных путей отображения матрицы меток состоит в использовании псевдоцветного индексного изображения.

В псевдоцветном изображении числа, которые идентифицируют каждый объект в матрице меток, отображаются разным цветом, который связан с соответствующей палитрой цветов. Такое представление позволяет выделять различные объекты (рисунок 1.5).

Для просмотра матрицы меток используется функция «label2rgb». Эта функция применяется при создании палитры, цвета фона и цвета каждого объекта изображения.

```
pseudo_color=label2rgb(labeled, @spring, 'c',
'shuffle');
imshow(pseudo_color);
```



Рисунок 1.5 – Представление матрицы меток в виде псевдоцветного изображения

Шаг 10: Измерение свойств объектов на изображении

Функция «regionprops» применяется для измерения свойств объектов в некоторой окрестности изображения и представляет результат в виде массива. Если применить это к изображению с отмеченными компонентами, то создается структура для каждой компоненты.

В этом примере используется функция «regionprops» для создания структурированного массива, содержащего некоторые основные свойства. Если установить свойствам параметр «basic», то функция «regionprops» возвращает три измеренных параметра: площадь, центроид (или центр масс) и ограничивающий прямоугольник (локальная окрестность). В данном случае ограничивающий прямоугольник представляет собой локальную окрестность, отображающую гранулы риса.

```
graindata = regionprops(labeled, 'basic')
```

Система MATLAB выдает такой результат

```
graindata = 101x1 struct array with fields:
Area
Centroid
BoundingBox
```

Для поиска области с 51-м отмеченным компонентом, необходимо найти поле «Areafield» и его 51 элемент в структуре массива «graindata».

```
graindata(51).Area
```

В этом случае результат будет таким

```
ans=140
```

Для поиска наименее возможного прямоугольника и центраоида (центра масс) некоторых компонентов используется следующий код:

```
graindata(51).BoundingBox,
graindata(51).Centroid
ans =
107.5000    4.5000    13.0000    20.0000
ans =
114.5000    15.4500
```

Шаг 11: Вычисление статистических свойств объектов изображения

Функции системы MATLAB могут использоваться для вычисления статистических свойств объектов. Сначала используется функция «max» для поиска наибольшего зерна. (В нашем примере наибольшими являются два зерна риса, которые соприкасаются.)

```
max([graindata.Area])
```

Результат представляется в виде

```
ans =404
```

Далее с помощью функции «find» найдем те компоненты, которыми отмечены зерна риса с этой площадью.

```
biggrain = find([graindata.Area]==404)
```

Результат представляется в виде

```
biggrain=59
```

Найдем средний размер всех зерен риса.

```
mean([graindata.Area])
```

Результат будет представлен в виде

```
ans =175.0396
```

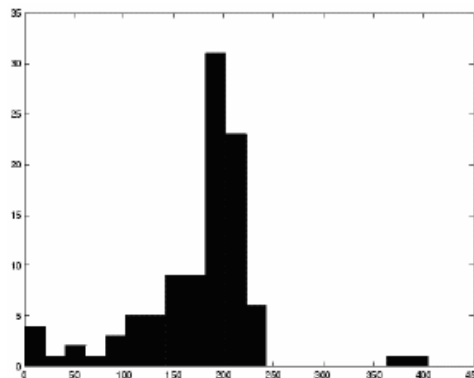


Рисунок 1.6 – Гистограмма распределения зерен по их размеру

Построим на рисунке 1.6 гистограмму распределения зерен риса по их размеру (при этом будем использовать 20 позиций при градации по оси площади). Из гистограммы видно, что основная часть зерен риса имеет площадь от 150 до 250 пикселей.

```
hist([graindata.Area],20)
```

Задания к работе

Написать программный модуль обработки изображения мазка крови и подсчета количества лейкоцитов или эритроцитов.

Контрольные вопросы

1. С использованием какой функции определяется число объектов на изображении? Каковы особенности данной функции?
2. Для чего анализируется матрица меток?
3. Каким образом измеряются объекты на изображении?
4. С помощью какой функции изображение загружается в Matlab? В каком формате изображенное изображение интерпретируется Matlab'ом?

Практическая работа №2

Анализ двумерных изображений в Matlab. Выделение границ

Анализ изображений

Применяя технику анализа, можно говорить о структуре изображений. Рассмотрим некоторые функции, которые используются при анализе изображений, а также их текстуры.

Выделение границ

Для определения границ объектов изображения можно использовать функцию «edge». В качестве одного из критериев для определения края используется резкий перепад интенсивностей пикселей изображения. Отметим, что некоторые методы выделения границ, в зависимости от своих особенностей, в большей степени выделяют вертикальные, горизонтальные или все границы одинаково.

Одним из наиболее эффективных методов выделения границ является метод Канни. Метод Канни отличается от других известных методов тем, что при определении границ использует два порога (для слабых и сильных границ). Слабые границы отмечаются в результирующем изображении только тогда, когда они соединены с сильными. Для зашумленных изображений данный метод обеспечивает наилучшее обнаружение границ по сравнению с остальными методами, но требует существенно большего времени.

Рассмотрим пример применения метода выделения границ Канни, а также сравним его с другим известным методом выделения границ:

Считаем и отобразим исходное изображение (рисунок 2.1).

```
I=imread('coins.png');  
imshow(I)
```



Рисунок 2.1 – Исходное изображение

Далее это изображение будет обработано операторами выделения границ Собела и Канни с последующей визуализацией (рисунок 2.2).

```
BW1 = edge(I, 'sobel');  
BW2 = edge(I, 'canny');  
imshow(BW1)  
figure, imshow(BW2)
```

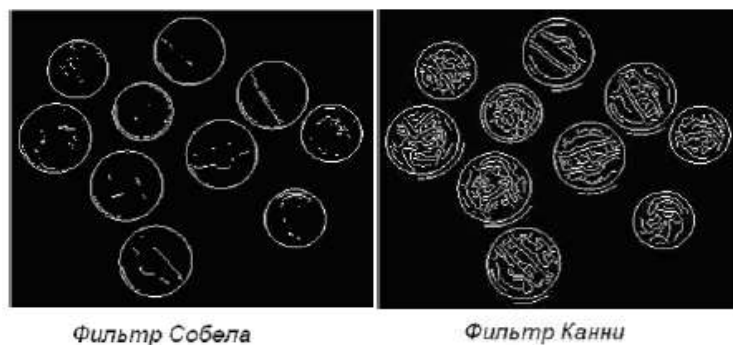


Рисунок 2.2 – Результат применения фильтров Собела и Канни к изображению

Для интерактивной демонстрации работы операторов выделения границ в приложении используется функция «`edgedemo`».

Отслеживание границ

Приложение включает две функции, которые могут быть использованы для поиска границ объектов на бинарном изображении. Это функции «`bwtraceboundary`» и «`bwboundaries`».

Функция «`bwtraceboundary`» возвращает координаты строк и столбцов всех пикселей, которые принадлежат границе объектов

изображения. Также необходимо описать начальную (стартовую) точку при отслеживании границ объектов изображения.

Функция «`bwboundaries`» также возвращает координаты сток и столбцов всех пикселей, которые принадлежат границе объектов изображений.

Для обеих функций ненулевые пиксели на бинарном изображении принадлежат объекту, а пиксели со значением 0 составляют фон.

Рассмотрим пример применения функции «`bwtraceboundary`» для отслеживания границ некоторого объекта бинарного изображения, а затем используем функцию «`bwboundaries`» для отслеживания границ всех объектов изображения (рисунок 2.3).

Считаем и визуализируем изображение.

```
I=imread('coins.png');  
imshow(I)
```

Преобразуем это изображение в бинарное. Функции «`bwtraceboundary`» и «`bwboundaries`» работают только с бинарными изображениями.

```
BW=im2bw(I); imshow(BW)
```

Определим координаты строки и столбца пикселя на границе объекта. Функция «`bwboundary`» использует эту точку в качестве начальной при отслеживании границ.

```
dim=size(BW)  
col=round(dim(2)/2)-90;  
row=min(find(BW(:,col)))
```

Функция «`bwtraceboundary`» выполняет отслеживание границ от начальной точки. В качестве обязательных аргументов необходимо ввести само бинарное изображение, координаты начальной точки и направление первого шага. В примере использовано северное направление («N»).

```
boundary=bwtraceboundary(BW,[row, col], 'N');
```


Отообразим сначала исходное полутоновое изображение, а потом координаты, которые возвращаются функцией «bwtraceboundary».

```
imshow(I)
hold on;
plot(boundary(:,2),boundary(:,1),'g','LineWidth',3);
```

Для отслеживания границ всех монет на изображении используется функция «bwboundaries». По умолчанию функция «bwboundaries» определяет границы всех объектов на изображении, включая вложенные объекты. В нашем примере на бинарном изображении некоторые монеты содержат темные области, которые функция «bwboundaries» интерпретирует как отдельные объекты. Для того чтобы функция «bwboundaries» отслеживала только границы монет, используется функция «imfill» для заполнения областей внутри каждой монеты.

```
BWfilled=imfill(BW,'holes');
boundaries=bwboundaries(BWfilled);
```

Функция «bwboundaries» возвращает массив значений, где каждая ячейка содержит координаты пикселей объекта изображения. Отообразим границы всех монет на исходном полутоновом изображении, используя координаты, возвращаемые функцией «bwboundaries».

```
for k=1:10
b=boundaries{k};
plot(b(:,2),b(:,1),'g','LineWidth',3);
end
```

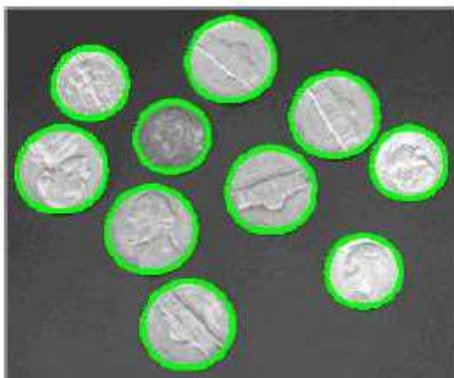


Рисунок 2.3 – Выделение границ на изображении

Задание

Выделить границы в любом медицинском изображении, заданные преподавателем, аналогично рассмотренному в теории.

Контрольные вопросы

1. Какие особенности метода Канни?
2. Какие функции Matlab используются при поиске границ на изображении?
3. Для чего используется функция «`bwboundaries`»?
4. Каковы особенности функции «`edge`»?

Практическая работа №3 Вейвлет анализ

Разложение сигнала по вейвлетам

Вейвлет-преобразование одномерного сигнала состоит в его разложении по базису, сконструированному из обладающей определенными свойствами функции (вейвлета) с помощью масштабных изменений и переносов. Каждая из функций этого базиса характеризует как определенную пространственную (временную) частоту, так и ее локализацию в физическом пространстве (времени):

$$s(t) = \sum_k C_k \psi_k(t),$$

где C_k - коэффициенты разложения, несущие информацию о сигнале, $\psi_k(t)$ - базисные функции.

Вейвлеты характеризуются своими временными и частотными образами. Временной образ определяется некоторой $\psi(t)$ - функцией, а частотный задается ее фурье-образом $\psi(\omega)$, который задает огибающую спектра вейвлета. В основе непрерывного вейвлет - преобразования лежит использование двух непрерывных и интегрируемых по всей числовой оси t или x функций: с нулевым значением интеграла 0 , определяющая детали сигнала и порождающая детализирующие коэффициенты; масштабирующая (или скейлинг-функция) с единичным значением интеграла, определяющая грубое приближение (аппроксимацию) сигнала и порождающая коэффициенты аппроксимации.

Функция $\psi(t)$ создается на основе той или иной базисной функции $\psi_0(t)$, которая, как и $\psi(t)$, определяет тип вейвлета:

$$\psi(t) = a^{-\frac{1}{2}} \psi_0\left(\frac{t-b}{a}\right),$$

где b – величина смещения по оси времени (координаты), a – коэффициент масштабирования.

Вейвлеты классифицируются по виду и особенностям базисной

функции $\psi_0(t)$, а также по имени ученого, впервые предложившего тот или иной вейвлет. Приведем примеры вейвлет-функций $\psi(t)$.

Пример 1. Вейвлет «мексиканская шляпа» (МНАТ-вейвлет) (рисунок 3.1, а):

$$\psi(t) = \left(\frac{2}{\sqrt{3}} \pi^{\frac{1}{4}} \right) (1 - t^2) \exp\left(-\frac{t^2}{2}\right).$$

Пример 2. Вейвлет Хаара (ортогональный вейвлет) (рисунок 3.1, б):

$$\psi(t) = \begin{cases} 1, & 0 \leq t \leq \frac{1}{2} \\ -1, & \frac{1}{2} < t < 1 \\ 0, & t < 0, t > 1 \end{cases}$$

Пример 3. Вейвлет Морле (Morlet) (рисунок 3.1, в):

$$\psi(t) = \exp\left(-\frac{t^2}{2}\right) \cos 95t.$$

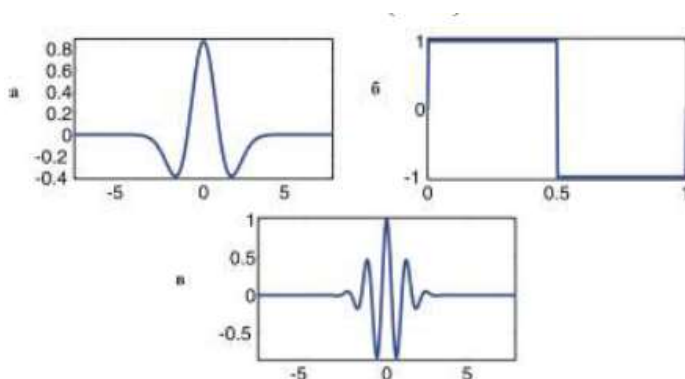


Рисунок 3.1 – Примеры вейвлетов: (а) «мексиканская шляпа», (б) вейвлет Хаара, (в) вейвлет – Морле

Прямое непрерывное вейвлет-преобразование сигнала $s(t)$, ограниченного в области \mathbb{R} , задается путем вычисления вейвлет-

коэффициентов:

$$W(a, b) = \int s(t) a^{\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) dt.$$

Обратное непрерывное вейвлет-преобразование осуществляется по формуле реконструкции во временной области, которая имеет ряд форм. Приведем эту формулу в виде использованном в пакете расширения системы Matlab – WaveletToolbox:

$$s(t) = \frac{1}{C_{\psi}} \iint W(a, b) a^{\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) \frac{dabd}{a^2},$$

где C_{ψ} - нормализующий коэффициент.

Энергетические характеристики сигналов

Полная энергия сигнала f может быть записана через амплитуды вейвлет-преобразования в виде:

$$E_f = \int f^2(t) dt = C_{\psi}^{-1} \iint W^2(a, b) \frac{dabd}{a^2}.$$

Плотность энергии сигнала $E_W(a, b) = W^2(a, b)$ - характеризует энергетические уровни исследуемого сигнала $f(t)$ в пространстве (a, b) =(масштаб, время).

Локальный спектр энергии позволяет проанализировать временную динамику передачи энергии процесса по масштабам – обмен энергией между составляющими процесс компонентами разного масштаба в любой заданный момент времени. Зная плотность энергии $E_W(a, b)$, можно с помощью окна определить локальную плотность энергии в точке b_0 (или t_0):

$$E_{\xi}(a, t_0) = \int E_W(a, b) \xi\left(\frac{b-t_0}{a}\right) db.$$

Оконная функция ξ «поддерживает» диапазон $0t$ и удовлетворяет равенству:

$$\int \xi(b)db = 1.$$

Если в качестве ξ выбрать функцию Дирака, то локальный спектр энергии примет вид:

$$E_{\delta}(a, t_0) = W^2(a, t_0).$$

Глобальный спектр энергии. Полная энергия распределена по масштабам в соответствии с глобальным спектром энергии коэффициентов вейвлет-преобразования:

$$E_W(a) = \int W^2(a, b)db = E_W(a, b)db.$$

Его называют также скалограммой или дисперсией вейвлет-преобразования.

Мера локальной перемежаемости:

$$I_W(a, t) = \frac{E_W(a, t)}{\langle E'_W(a, t) \rangle_t}.$$

- мера локальных отклонений от среднего поля спектров на каждом масштабе: она позволяет определить степень неравномерности распределения энергии по масштабам (угловыми скобками здесь обозначено усреднение).

Мера контрастности:

$$C_W(a, t) = \frac{E_W(a, t)}{E'_W(a, t)},$$

$$E'_W(a, t) = \int_{a'=0}^{a'=a} E_W(a', t)da'.$$

позволяет определять даже самые малые изменения в сигнале, когда

необходимо, например, выявить структурированность слабого сигнала или слабые вибрации на фоне крупной структуры.

Вейвлет-анализ сигналов с помощью картин значений коэффициентов вейвлет-преобразований

Картина значений вейвлет-коэффициентов представляет значение коэффициентов вейвлетов в плоскости масштаб (номера коэффициентов) - время (или пространственная координата). Внизу картины значений вейвлет-коэффициентов расположены коэффициенты с малыми номерами, дающие детальную картину сигнала, а сверху - с большими номерами, дающие огрубленную картину сигнала. Результатом вейвлет-анализа заданного сигнала с помощью картин вейвлет-коэффициентов являются графики:

- 1) график функции заданного сигнала;
- 2) картины значений вейвлет-коэффициентов;
- 3) проекции картины значений вейвлет-коэффициентов при фиксированном значении коэффициента a ;
- 4) скелетон (картина линий локальных экстремумов) картины значений коэффициентов вейвлетов.

Для построения картин значений коэффициентов вейвлет-преобразований сигналов используется вычислительный пакет WaveToolbox среды MATLAB, который запускается командой `wavemenu`. В результате появится окно со списком разделов вейвлет-преобразований (рисунок 3.1).

Для получения справки по какому-либо типу вейвлета при работе в командном режиме MATLAB достаточно исполнить команду `waveinfo('type')`, указав тип вейвлета. Для просмотра же вейвлетов достаточно исполнить команду `wavemenu` и в появившемся окне со списком разделов ВП нажать кнопку `WaveletDisplay`. Нажатие этой кнопки выводит окно просмотра вейвлетов, в котором имеется возможность просмотра: общей информации о вейвлетах, выбранного вейвлета (с именем «Name») и информации о нем.

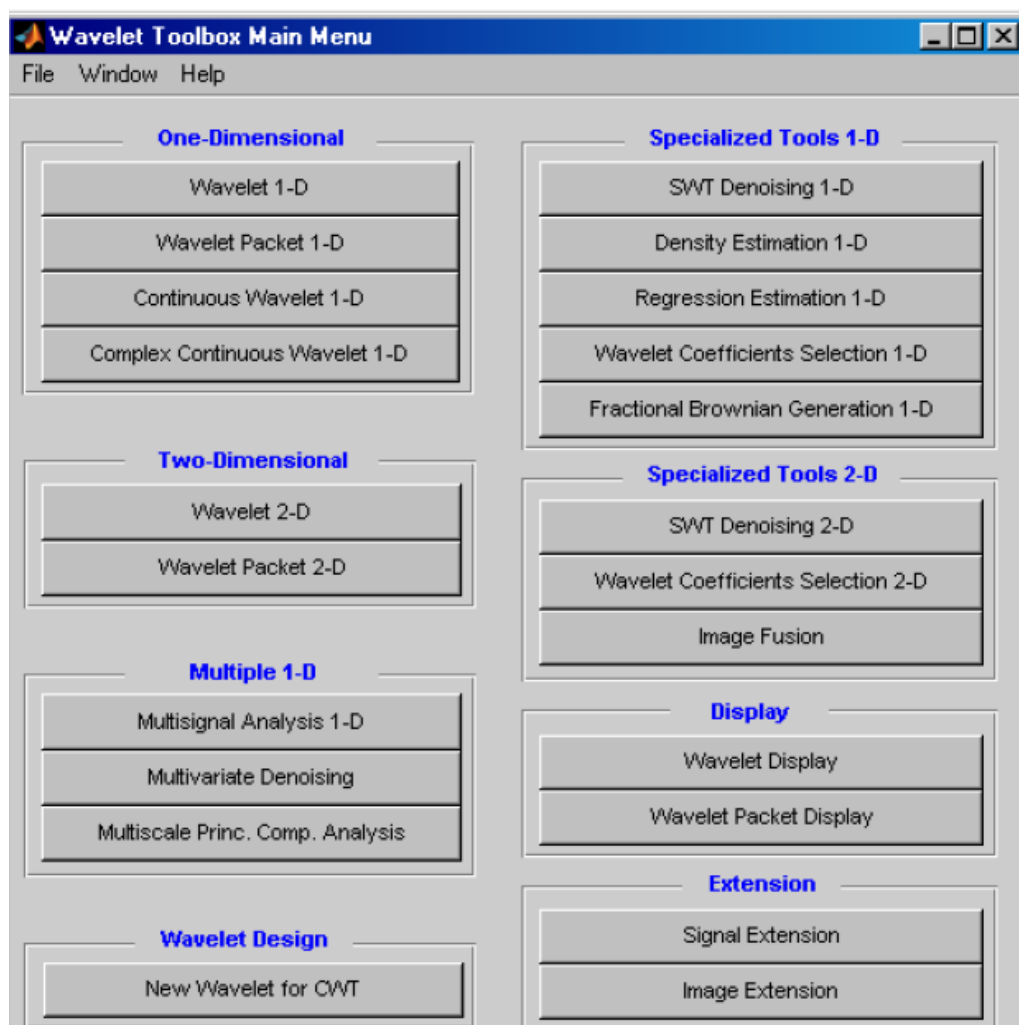


Рисунок 3.1 – Главное меню вычислительного цикла WaveToolbox в Matlab

При вейвлет-анализе заданного сигнала можно использовать набор встроенных вейвлетообразующих функций (на панели справа меню Continuouswavelet 1-D) или задать необходимый вейвлет с помощью инструмента вейвлет-менеджер - wavemngr. Для просмотра встроенных вейвлетов воспользуйтесь кнопкой WaveletDisplay в окне главного меню WaveletToolboxMainMenu.

Для анализа тестовых сигналов с помощью картин значений вейвлет-коэффициентов выберите кнопку главного меню Continuouswavelet 1-D. Появится окно, в котором производится вейвлет-анализ заданного сигнала с помощью непрерывных вейвлетов.

Задания тестовых сигналов производится следующим образом:
Нажмите

- file в меню Continuous wavelet 1-D;

- выберите Loadsignal;
- укажите путь к файлу, в котором задается тестовый сигнал. Файл с заданным сигналом должен быть сохранен в mat-формате.

В Matlab вейвлет-коэффициенты $W(a, b)$ обозначены $C(a, b)$.

Пример 1. Проведем вейвлет-анализ с помощью картин вейвлет-коэффициентов для сигнала

$X_k = \frac{2}{\sqrt{k+1}} \cos\left(2\pi \frac{k}{T/\sqrt[k+1]} + \frac{\pi}{3}\right)$, где параметры сигнала $k=0, 1, \dots, 255$ на основе вейвлета Хаара.

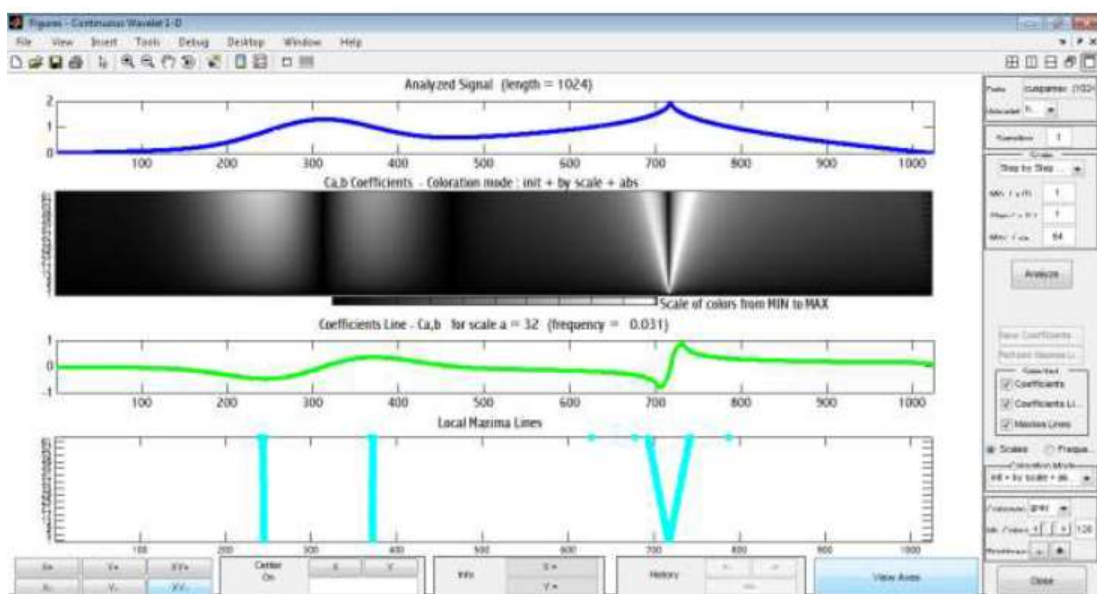


Рисунок 3.2 – Вейвлет-анализ на основе вейвлетообразующей функции Хаара тестового сигнала

Локальным особенностям (нарушениям гладкости) на картине вейвлет-коэффициентов отвечают вертикальные полосы, выходящие из точки, где находится особенность (рисунок 3.2). Пикам сигналов соответствует сгущение светлых областей вейвлет-картин значений коэффициентов, а впадинам сгущение темных полос. Чем резче выражена особенность сигнала, тем сильнее она выделяется на картине значений вейвлет-коэффициентов. Анализ тестовых сигналов с использованием комплексных непрерывных вейвлетов производится в окне главного меню: ComplexContinuouswavelet 1-D (рисунок 3.1). Здесь используются два метода анализа заданного сигнала по модулю $(A = \text{Re}^2 + \text{Im}^2)$ и углу $\varphi = \text{arctg}(\text{Im}/\text{Re})$, где Im и Re - соответственно мнимая и

действительная часть анализируемой функции.

В приводимых ниже примерах прямое ВП реализуется функцией «cwt», которая может быть представлена в нескольких формах, например: `COEF=cwt(S, SCALES, «wname» PLOTMODE, XLIM)`, где строка `S`-сигнал, строка `SCALES` - задание диапазона и шага изменения параметра, а строка «wname» - имя (тип) вейвлета, строка `PLOTMODE` - настройка цвета: «lvl» - окраска шаг за шагом, «glb» - окраска с учетом всех коэффициентов, «abslvl» или «lvlabs» - окраска шаг за шагом с использованием абсолютных значений коэффициентов, строка `XLIM` - строка переменных настройки.

Пример 2. Гармоническое колебание

```
function garm
t=0:0.00001:0.0004; A1=1; F1=10000; a1=45;
s=A1*soc(2*pi*F1*t+a1);
figure(1); plot(t, s); axis([0 0.0004 -3 3]);
grid on;
subplot(211), plot(t, s); title('Сигнал S(t)')
subplot(212), c = cwt(s, 1:2:32, 'mexh', 'abslvl',
[0 10]);
title('Вейвлет-спектр'); xlabel('Временной сдвиг,
b'); ylabel('Временной масштаб, a');
end
```

И хотя спектрограмма гармонического колебания, представленная на рисунок 3.3, особой выразительностью не отличается, на ней отчетливо видны переходы сигнала через нуль (темный тон) и экстремальные точки (светлый тон).

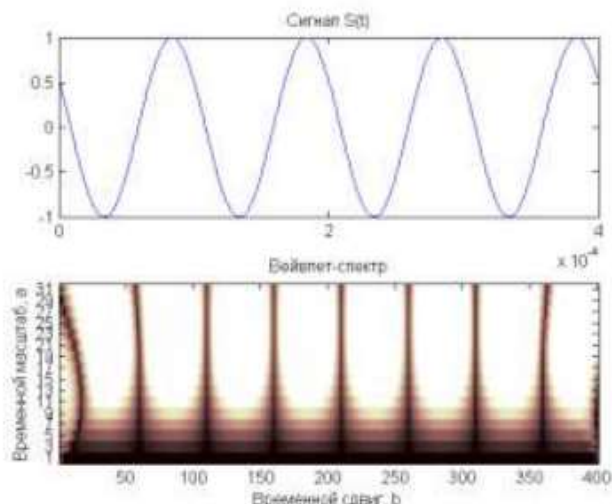


Рисунок 3.3 – Спектрограмма гармонического колебания

Численный расчет энергетических характеристик сигналов

Для вычисления энергетических характеристик заданного сигнала нужно записать формулу энергетических характеристик в виде конечных сумм в вычислительном пакете Matlab и отобразить результаты в виде графиков.

Запишем интегральные формулы в виде сумм:

-глобальный спектр энергии:

$$E_w = \sum_{b=0}^{K-1} |W(a, b)|^2;$$

-мера локальной перемежаемости:

$$I_w = \frac{K |W(a, b)|^2}{\sum_{b=0}^{K-1} |W(a, b)|^2};$$

-мера контрастности:

$$C_w = \frac{K |W(a = p, b)|^2}{\sum_{a=1}^p |W(a, b)|^2}$$

Пример 3. Приведем пример построения графиков основных энергетических характеристик (рисунок 3.4) для сигнала, заданного функцией Вейерштрасса со следующими параметрами $\sigma = 3,3$; $b = 2,5$; $s = 0,005$; $D = 1,5$; $N = 2$; $K = 255$, случайная фаза ψ_n распределена равномерно на отрезке $[0, 2\pi]$ (рисунок 3.3). Для нахождения функции $\psi_{a,n}$, использовалась вейвлетообразующая функция «мексиканская шляпа».

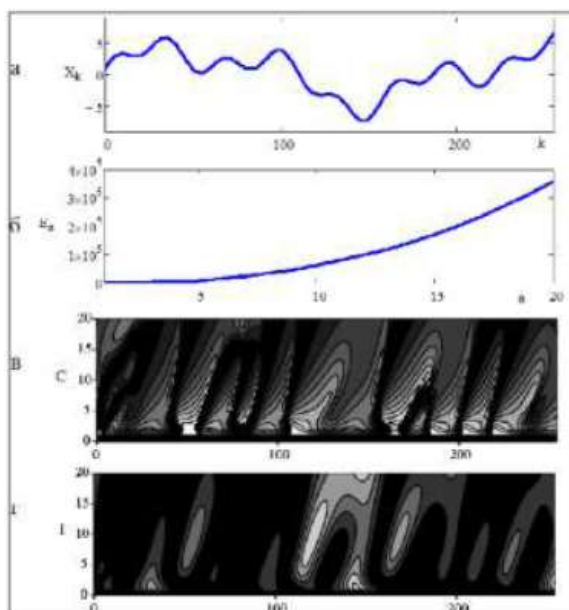


Рисунок 3.4 – График основных энергетических характеристик для сигнала, заданного функцией Вейерштрасса

Задания

1. Сформировать сигнал в виде прямоугольного импульса с шумом. Построить график сигнала и вейвлет-спектра. В качестве вейвлета использовать `mat`.

2. Провести вейвлет-анализ с помощью картин вейвлет-коэффициентов для сигналов, заданных функцией Вейерштрасса с параметрами $N=6, D=1,8$ и $N=6, D=1,2$. Построение проекций картин значений вейвлет-коэффициентов осуществлять при фиксированных значениях коэффициента $a=8, 12$. Объясните различия в структуре сигналов. Провести анализ на основе вейвлета «мексиканская шляпа».

3. Определить глобальный спектр энергии вейвлет-преобразования функции Вейерштрасса с параметрами $N=6, D=1,2$ и дать ему объяснение. Определить меру локальной перемежаемости для заданного сигнала.

Задание сигналов в `wavemenu` вычислительного пакета `WaveToolbox` среды `MATLAB` осуществляется с помощью файлов с расширением `mat`.

Для создания `mat`-файла нужно воспользоваться следующей процедурой:

```
savefile = 'test001.mat';  
Z=математическая формула сигнала;  
save(savefile, 'Z')
```

В результате создается файл с именем test001 и расширением mat, в который будет записан сигнал Z.

Контрольные вопросы:

1. Как производится разложение сигнала по вейвлетам?
2. Приведите примеры вейвлетообразующих функций.
3. Что позволяют определить меры контрастности и локальной перемежаемости?
4. Приведите определение скалограммы.
5. Как определяются локальный и глобальный спектр энергии?
6. Что такое скелетон вейвлет-преобразования сигнала?
7. В каких случаях эффективно применение вейвлет-преобразований?