

ОБРАБОТКА ИЗОБРАЖЕНИЙ ДЛЯ ОПРЕДЕЛЕНИЯ КОЛЛЕКТОРСКИХ СВОЙСТВ ПОРОД В GNU OCTAVE (MATLAB)

Цель работы: ознакомление с методическими приемами анализа микроизображений горных пород в средах научного программирования.

Задание: провести анализ изображений в соответствии с вариантом с применением приводимых методик. Применить фильтры изображения, описать достигаемый эффект. Бинаризовать изображение, выделить кластеры и провести их простейший анализ.

Подготовить содержательный отчет, сделать выводы.

Примечание. Для выполнения настоящей работы используется бесплатная программа GNU Octave 4.0 (<http://www.octave.org>), имеющая существенную совместимость с Matlab. Для работы программы обработки изображений в Octave, требуется скачать модуль *image* (по [ссылке](#)), а затем установить его с помощью консольной команды Octave:

```
pkg install C:\my_folder\image-2.4.1.tar.gz
```

Здесь `C:\my_folder\` – папка, в которой находится скачанный архив с модулем *image*, `image-2.4.1.tar.gz` – имя файла архива. Перед выполнением программы с применением модуля его необходимо загрузить командой *pkg load all*.

Внимание! Некоторые коды примеров могут содержать небольшие неточности, препятствующие их корректному выполнению. Это означает, что необходимо найти и исправить «неточность», что требует понимания работы кода программы. *Разумеется*, предоставляемый по итогам работы отчет должен содержать корректный код программы, без ошибок, с которыми программа работать не сможет.

6.1. Коллекторские свойства горных пород

По своему происхождению горные породы могут быть разделены на осадочные, магматические и метаморфические.

Для прогноза, поисков и эксплуатации нефтегазовых месторождений, проектирования подземных хранилищ природного газа бывает важно определить, является ли порода горная порода коллектором (т.е. может вмещать жидкость или газ) или флюидоупором (т.е. не способна вмещать жидкость или газ).

В качестве главных свойств пород, отвечающих за их коллекторские качества выступают *пористость* и *проницаемость*.

Пористость горной породы φ выражается как отношение объема пор $V_{\text{пор}}$ к объему образца $V_{\text{обр}}$ и может быть выражена в %:

$$\varphi = \frac{V_{\text{пор}}}{V_{\text{обр}}} \quad (1)$$

Пористость соответствует емкости породы – т.е. ее способность вмещать флюид – нефть, газ или природные воды. Пористость также может быть

выражена как функция давления, так как при увеличении давления пористость уменьшается за счет сокращения объема пор. В практике геологических исследований обычно применяют лабораторные методы измерения пористости.

Однако, в случае однородного распределения пор, каналов или трещин в породе, пористость может определена в результате анализа микрофотоизображения породы – *шлифа* (рис. 6.1).

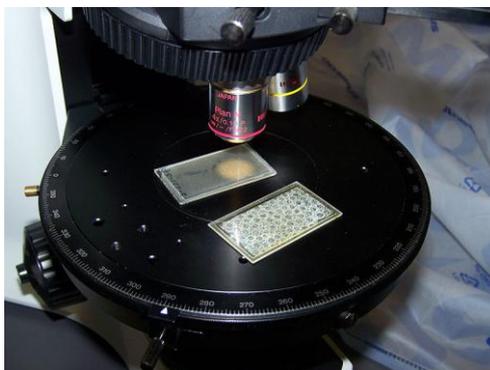


Рис. 6.1. Шлифы на предметном столике микроскопа

Шлифом называется тонкая (доли миллиметра) полированная пластинка породы, которая приклеивается к препаровальному стеклу с помощью канадского бальзама (прозрачная смесь смол хвойных растений).

Изучение шлифов и их описание выполняется в специальных оптических приборах, называемых поляризационными микроскопами (рис. 6.2).



Рис. 6.2. Поляризационный микроскоп Nikon Eclipse E200POL

Для приближенного расчета пористости при анализе плоского изображения можно заменить отношение объема пор

Проницаемость горной породы рассматривается неотрывно от течения в ней жидкости и может быть выражена с помощью отношения Кармана-Козени [1]:

$$K = \frac{1}{8\tau A_p^2} \frac{\varphi^3}{(1-\varphi)^2} \quad (2)$$

Проницаемость зависит от текстуры породы, выраженной в качестве

показателя кривизны τ и удельной площадью поверхности частиц породы A_v .

6.2. Методы морфологического анализа изображений в Octave (Matlab)

Для ознакомления с методами обработки изображений в Octave рассмотрим следующий пример. Для занятия необходимо использовать изображение в папке files архива лабораторной работы.

Внимание! Для корректной работы Octave имя файла и путь к нему не должны содержать кириллических символов.

Рассмотрим изображение неких изометрических (округлых) объектов, находящихся на плоской поверхности (рис. 6.3).

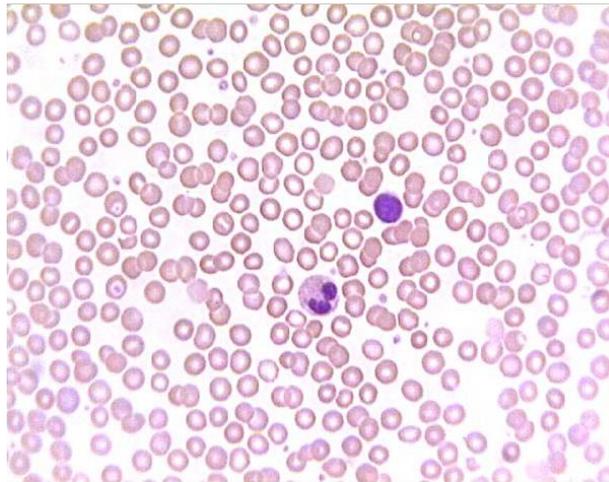


Рис. 6.3. Изображение клеток крови, предмет анализа в примерах (с сайта <http://habrahabr.ru>)

Что можно понять на первый взгляд смотря на эти объекты? Прежде всего то, что приемы анализа должны зависеть от сущности наблюдаемых объектов. Некоторые объекты на рисунке *circles_foto.png* содержат внутри светлую область. Это может быть или пустота или блик – т.е. засветка фото, вызванная более интенсивным отражением определенных частей изображения. Если это блик, а не пространство, то, например, для задач учета площади объектов нам необходимо заполнить его пикселями. Как это сделать, будет рассмотрено в примерах этой лабораторной работы.

Бинаризация – преобразование цветного (RGB) изображения в логическое, матрица которого состоит из 0 и 1. Это бывает нужно для выделения полезного сигнала, т.е. необходимой для текущего исследования части изображения.

Пример 1. Открытие изображения и его бинаризация. Обратите внимание на комментарии в тексте программы

```
clear; %очистка памяти

pkg load all; %загружаем модули, только для Octave!

A=imread('circles_foto.png'); %загрузка цветного изображения

A_gray=rgb2gray(A); %преобразование изображения в оттенки серого

imshow(A_gray); %вывод серого изображения
figure; %создание нового графического окна

A_bw=(A_gray<180); %бинаризация изображения

imshow(A_bw); %вывод бинаризованного изображения в графическом окне
```

В результате выполнения программы будет выведено два графического изображения, серого A_{gray} и черно-белого (логического) A_{bw} . Принципиально важно понять, как в примере изображение было бинаризовано:

```
A_bw=(A_gray<180); %бинаризация изображения
```

Запись $A_{gray}<180$ означает, что мы выбираем пиксели изображения A_{gray} , яркость которых имеет значения меньше **180** и рассматриваем их как логическую 1 для нового изображения A_{bw} . Логические 1 показываются функцией `imshow()` как белый цвет, 0 как черный цвет. Напомню, что яркость изображения 255 – это белый цвет (фон) исследуемой фотографии. Понятно, что в выборку $A_{gray}<180$ попадут самые и умеренно темные участки изображения (рис. 6.4а). А если сделать выбор $A_{gray}<120$, то мы выберем уже наиболее темные объекты, а остальные будут игнорироваться (рис. 6.4б).

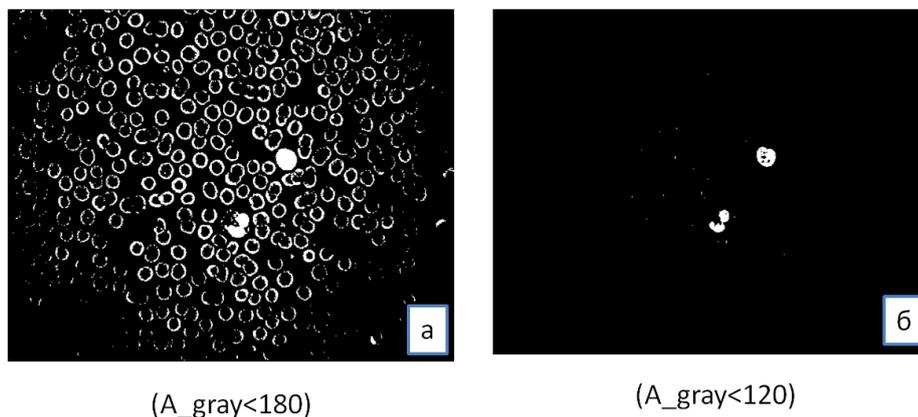


Рис. 6.4. Результат формирования бинарного изображения с различным порогом бинаризации. а – порог бинаризации 180; б – порог бинаризации 120. Светлые части изображения соответствуют логической 1.

Изменяющееся значение предельного значения оттенка серого называется *порогом бинаризации*. Порог бинаризации подбирается для каждой фотографии экспериментально.

После бинаризации исходного изображения заметно, что теряются

внутренние части объектов. Необходимо найти способ их заполнить. Для этой цели нам помогут следующие морфологические операции логических изображений.

Морфологическая эрозия – морфологическое удаление пикселей, границ объектов логического изображения, равных 1 с помощью «стирателя» – структурного элемента, особой матрицы, создаваемого функцией `strel(n)`, где `n` – размерность объекта.

Морфологическая дилатансия – морфологическое наращивание пикселей, прилегающих к объектам с помощью аналогичного структурного элемента.

Пример 2. Заполнение внутренних частей объектов изображения.

Внимательное рассмотрение бинарного изображения из примера №1 позволяет сделать вывод, что существенная часть объектов изображения потеряна.

Ввиду того, что *Matlab* и *Octave* не вполне совместимы, приводим код для них отдельно:

Код *Octave* (*Matlab*):

```
A_bw2 = edge(A_gray,'canny', 0.2, 2); %применяем алгоритм Canny для
%поиска границ на изображении

%создаем структурный элемент, прямоугольник 2x2, которым будем
%воздействовать на изображение
se = strel('square',2);

%выполнение операции дилатансии,
%т.е. морфологического наращивания элементом se
A_bw2=imdilate(A_bw2,se);

%заполнение пустот на изображении
A_bw3 = bwfill( A_bw2, 'holes',8);

%создаем структурный элемент большего размера
se = strel('square',3);

%убираем имеющийся на изображении пиксельный "мусор"
%морфологической эрозией
A_bw3=imerode(A_bw3,se);

%выводим результат обработки изображения
imshow(A_bw3);
```

Код *Matlab* (в *Octave 4* не выполняется):

```
%применяем алгоритм Canny для
%поиска границ на изображении
A_bw2 = edge(A_gray,'canny', 0.15, 2);
A_bw2 = imfill(A_bw2,'holes');
%strel - структурный элемент
se = strel('disk',1);

%imopen морфологическое открытие, эрозия после дилатансии
A_bw3 = imopen(A_bw2,se);
imshow(A_bw3);
```

В результате применения кода, показанного выше, будет получено

бинарное изображение A_{bw3} , аналогичное (рис. 6.5).

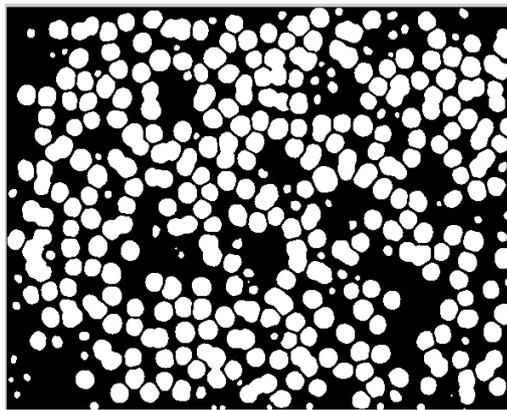


Рис. 6.5. Результат комбинации действий, направленных на выделение краев, заполнение границ, эрозии и дилатансии.

Рассмотрение изображения (см. рис. 6.5) показывает, что цели преобразования изображения достигнуты. Выделена большая часть объектов, заполнены пустоты, обусловленные бликом. Специальная функция выделения кластеров бинарного изображения $bwlabel()$ возвращает в качестве результата матрицу подписанных объектов и их количество.

Знание масштаба изображения (соответствие пикселя, например, мм^2) позволит нам рассчитать площадь объектов.

Пример 3. Функция $bwlabel$

Эта функция позволяет на основе анализа бинарного (логического) изображения получить матрицу подписанных кластеров (взаимосвязанных точек) и их количество (число объектов):

```
[B,L] = bwlabel(A_bw2);  
%B-матрица нумерованных кластеров, L-их число  
imagesc(B); %это функция показа матрицы, а не изображения  
title('Regions matrix');  
disp('Number of regions=');  
disp(L); %вывод в командном окне количества кластеров
```

В результате выполнения кода выше будет выведено количество объектов на изображении и матрица пронумерованных кластеров (рис. 6.6)

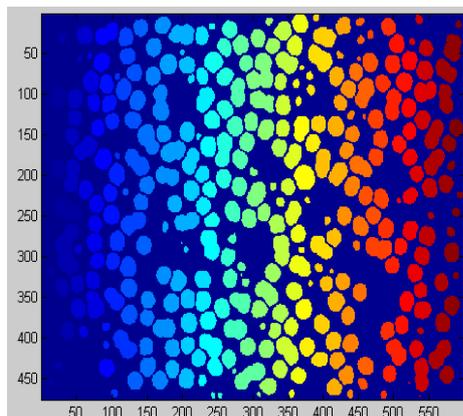


Рис. 6.6. Матрица кластеров объектов – результат выполнения функции

bwlabel(). Объекты показаны цветами, в соответствии с номером

Теперь можно использовать функцию *regionprops()*, (от англ. «region properties», свойства региона) для вычисления свойств объектов. Эта функция может вычислить многие свойства, но нам потребуется только два: 'Centroid' (координаты центра), 'Area' (площадь кластера в пикселях). Свойства передаются особому объекту типа Structure («структура»), который может их хранить.

Пример 4. Функция *regionprops*

Эта функция позволяет на основе анализа бинарного (логического) изображения получить матрицу подписанных кластеров (взаимосвязанных точек) и их количество (число объектов):

```
stats = regionprops(B, 'Centroid', 'Area');

imshow(A);
hold on; %это значит, следующий график будет поверх картинки

xc=[]; %пустые массивы для координат центров
yc=[];

%пустая переменная для хранения суммарной площади
sum_area=0;

for k = 1:L %обход объекта stats

    xc = [xc stats(k).Centroid(1)]; %добавляем координаты центра в
их векторы xc и yc
    yc = [yc stats(k).Centroid(2)];

    sum_area=sum_area+stats(k).Area; %добавляем площадь k-того
объекта к sum_area

end
%выводим график координат центра поверх
plot(xc,yc, '+');

disp('Total area=');
disp(sum_area);
```

Выполнение этого кода выведет нам координаты центра обработанных объектов поверх исходной картинки и общую площадь в командном окне.

Пример 5. Инвертирование

Иногда полезным является инвертирование изображений. Операция инвертирования ~ превращает 1 изображения в 0 и наоборот. Так, вызов:
В командном окне:

```
A= [0 1 0]
A =

     0     1     0

>> ~A
ans =

     1     0     1
```

Рекомендации.

- Параметры приводимых функций могут изменяться в зависимости от ситуации, сжатый объем методических указаний не позволил привести их здесь вполне. Читателю рекомендуется обратиться к встроенной в программу справке и материалам сети Интернет;

- В Matlab/Octave для вывода строковых сообщений используются следующие функции. Преобразование числа в строку, выполняемое в Scilab функцией *string(Число)*, в Matlab/Octave требует применения функции *num2str(Число)*. Объединение строк, осуществляемое в Scilab оператором сложения, в Matlab/Octave требует *strcat(Строка1, Строка2, ...Строка n)*;

- Существуют и другие несовместимости. При возникновении технических затруднений, необходимо читать справку по функциям, применяемым в Matlab/GNU Octave;

- Прервать выполнение программы можно, нажав в командном окне комбинацию клавиш «Ctrl+C»;

- **Вариантов меньше, чем студентов. Однако, при самостоятельном характере работы, повторяющиеся решения исключены. Поэтому, работы, содержащие однотипное решение, будут отклоняться. Засчитываться будут работы, присланные раньше.**

Внимание! Начало координат изображения в Matlab/Scilab/Octave – левый верхний угол. Столбцы и строки графического изображения развернуты на 90 относительно его матричного представления.

Задания для самостоятельной работы

- Проверить код Matlab, найдя ошибки, препятствующие его нормальной работе (если таковые имеются);

- Составить глоссарий применяемых в уроке функций Matlab, с кратким описанием;

- Используя материалы примеров, в зависимости от варианта, **рассчитать или пористость образца или количество объектов на микрофотографии** (Приложение А) (подпапка *variants* папки *files*). Необходимо учитывать, что поры в породе могут быть заполнены, а могут быть и не заполнены нефтью. Вне зависимости от этого, поры подлежат учету.

Требования к отчету

Отчет о выполнении лабораторной работы должен содержать:

- тему лабораторной работы на титульной странице;
- автособираемое оглавление, с номерам страниц;
- необходимые теоретические сведения по теме решаемых задач, вставка рисунков из методички должна быть обоснованной;
- исходные данные и последовательность их обработки (преобразования);
- поэтапный расчет в виде вставок кода Scilab/Matlab в объектах «Подпись»;
- результаты выполнения кода Scilab/Matlab в виде вывода в командном окне;
- графики, выполненные в Scilab/Matlab, оформленные как рисунки, после ссылок на них в тексте отчета;
- интерпретацию результатов работы моделей;
- отметку преподавателя о выполнении лабораторной работы.

Рекомендуемая литература

1. An Introduction to Reservoir Simulation Using MATLAB. User Guide for the Matlab Reservoir Simulation Toolbox (MRST) / Department of Applied Mathematics Oslo, Norway. SINTEF ICT, 2014. 213 p.
2. Алексеев Е.Р. , Чеснокова О.В. Введение в Octave для инженеров и математиков: / Е.Р. Алексеев, О.В.Чеснокова М.: ALT Linux, 2012. 368 с.
3. Материалы по продуктам MATLAB & Toolboxes // [Электронный ресурс]: Математический сайт Exponenta.ru. Веб-сайт. URL: <http://matlab.exponenta.ru/index.php> (Дата обращения: 05.11.2015)

Варианты исходных данных для решения самостоятельной задачи,
п – поры, о – объекты. Номер варианта соответствует номеру файла jpg

Вар	Описание	Примечание
1	Высокопористый водорослевый известняк. Глубина 1700 м. Материал – багряные водоросли. Поры – «прозрачные области изображения»	п
2	Мрамор с керитом (битуминозное вещество). Поляризованный свет. Поры – «темные области изображения»	п
3	Материал – пенобетон. Рассчитать пористость по наиболее контрастной части изображения	п
4	Трещиноватый нефтеносный известняк. Определить пористость (трещиноватость) породы	п
5	Комковатый известняк с вторичным цементом. Битум, заполняющий поры – темно-коричневый.	п
6	Перекристаллизованный известняк битуминозный. Битум темно-коричневого цвета заполняет поры	п
7	Пемза вулканическая	п
8	Золотины из россыпи р. Анабар.	о
9	Раковинки фораминифер – морских одноклеточных простейших	о
10	Трещиноватый металл. Разрушенный коррозией материал	п
11	Трещиноватый металл. Разрушенный усталостью материал	п
12	Битуминозный песчаник. Битум – темно-коричневый	п