

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



**ДОСЛІДЖЕННЯ МЕТОДІВ ОБРОБКИ
ЗОБРАЖЕНЬ В MATLAB**

**Методичні вказівки
до лабораторної роботи № 3**

**з курсу “ Системи запису та відтворення інформації ”
для студентів базового напрямку 6.170102
“Системи технічного захисту інформації”**

*Затверджено
на засіданні кафедри
"Захист інформації"
Протокол № від ..2009 р.*

Львів 2009

Дослідження методів обробки зображень в **MatLab**: Методичні вказівки до лабораторної роботи № 3 з курсу "Системи запису та відтворення інформації" для студентів базового напрямку "6.170102 "Системи технічного захисту інформації" / Укл. *Я. Р. Совин* - Львів: Національний університет "Львівська політехніка", 2009. - 8 с.

Укладачі:

Я. Р. Совин, к.т.н., доцент

Відповідальний за випуск:

Рецензент:

Мета роботи – ознайомитися з основними можливостями обробки зображень реалізованими в середовищі **MatLab**.

1. ТЕОРЕТИЧНІ ВІДОМОСТІ

Основним поняттям цифрового зображення є піксел (англ. pixel – picture elements). Піксел – це маленький прямокутник (майже квадрат), що має координати і певні атрибути. Такими атрибутами можуть бути колір пікселя, його прозорість і т.д.

Зображення можуть бути наступних типів:

Бінарні (чорно-біле) зображення задається матрицею, елементи (піксели) якої приймають два значення 0 (білий) і 1 (чорний).

Напівтонові (англ. grayscale) зображення задаються матрицею, елементи якої приймають дійсні значення від 0 до 1 і характеризують яскравість відповідного пікселу – відтінок сірого кольору.

Палітрове зображення задається двома матрицями X і map . Карта кольорів map , (палітра) складається з трьох стовпців, які задають інтенсивність червоного (R), зеленого (G) і синього (B) кольорів в межах від 0 до 1. Значення елементу матриці X є номер рядка в матриці map , в якому вказані інтенсивності кольорів пікселу, що відповідають елементу матриці X .

RGB-зображення представляє собою тривимірний масив розміром $n \times m \times 3$. Ці три ($n \times m$)-матриці, містять інтенсивності червоного (R), зеленого (G) та синього (B) кольорів кожного пікселу. Найбільш часто використовують кольорові зображення, що мають 16, 256, 65536 (High Color) і 10.7 мільйонів (True Color) кольорів.

Щоб завантажити зображення в робочу область MatLab можна використати функцію **imread**:

$A = \text{imread}(\text{filename}, \text{fmt})$ або $[X, map] = \text{imread}(\text{filename}, \text{fmt})$

Ця функція читає з файлу з іменем `filename` напівтонове або кольорове зображення у масив A . Якщо зображення напівтонове, то A – двовимірний масив, якщо зображення кольорове, то A – тривимірний масив розміром $n \times m \times 3$. У другому форматі цієї функції з файлу з іменем `filename` читається палітрове зображення в масив A з картою кольорів map . Параметри `filename` і `fmt` є масивами символів. Нижче приведені допустимі значення параметру `fmt`:

Значення параметру <code>fmt</code>	Назва формату
'bmp'	Windows Bitmap (BMP)
'cur'	Windows Cursor resources (CUR)
'hdf'	Hierarchical Data Format (HDF)
'ico'	Windows Icon resources (ICO)
'jpg' або 'jpeg'	Joint Photographic Experts Group (JPEG)
'png'	Portable Network Graphics (PNG)
'tif' або 'tiff'	Tagged Image File Format (TIFF)
'pcx'	Windows Paintbrush (PCX)
'gif'	Graphics Interchange Format (GIF)
'xwd'	X Windows Dump (XWD)

Для запису масиву з зображенням у файл служить функція **imwrite**. Вона має наступні форми:

imwrite(A , `filename`, `fmt`)

записує зображення у файл з іменем `filename` у форматі `fmt` з масиву A . A може бути матрицею розміром $n \times m$ для напівтонового зображення і масивом $n \times m \times 3$ для кольорового зображення. При цьому числа з плаваючою комою (типу `double`) в діапазоні $[0, 1]$ перетворюються у 8-бітні цілі числа в діапазоні $[0, 255]$.

imwrite(X, map, filename, fmt)

записує па літрове зображення у файл з іменем filename у форматі fmt з масиву X і відповідної карти кольорів map.

imwrite(..., filename)

аналогічна попереднім функціям, але формат файлу визначається за розширенням filename.

Одержати інформацію про графічний файл дозволяє функція

info = **imfinfo**(filename, fmt)

повертає структуру info, в полях якої міститься інформація про зображення в графічному файлі з іменем filename і форматі fmt. В структурі info кількість полів залежить від даного файлу і його формату. Проте перші дев'ять основних полів завжди однакові для всіх типів файлів:

Ім'я поля	Тип даних	Опис
Filename	Рядок	Ім'я файлу, якщо файл знаходиться в біжучому каталозі, або повний шлях до файлу.
FileModDate	Рядок	Дата і час останньої модифікації файлу.
FileSize	Число	Розмір файлу в байтах.
Format	Рядок	Формат файлу, тобто параметр fmt.
FormatVersion	Рядок або число	Версія формату.
Width	Число	Ширина зображення в пікселях.
Height	Число	Висота зображення в пікселях.
BitDepth	Число	Глибина кольору зображення в бітах на піксел.
ColorType	Рядок	Тип зображення: 'truecolor' для RGB-зображення, 'grayscale' для напівтонового зображення, 'indexed' для палітрового зображення.

Для перетворення палітрового зображення в напівтонове використовується функція

I = ind2gray(X, map).

Для перетворення кольорового RGB-зображення в напівтонове використовується функція

I = rgb2gray(RGB).

Для перетворення матриці A в напівтонове зображення I служить функція

I = mat2gray(A),

яка здійснює масштабування елементів матриці A до інтервалу від 0 до 1.

1.2. Функції MatLab для обробки зображень

Для дослідження методів видалення шуму з зображення потрібні тестові зображення з шумовими компонентами. Накладання шуму на зображення можна здійснити з допомогою функції imnoise:

J = imnoise(I, type)

додає до зображення I сигнал з вказаним типом шуму:

'gaussian' - гаусівський білий шум;

'salt & pepper' - шум типу 'сіль і перець' у вигляді включених або виключених пікселів;

'speckle' - мультиплікативний шум.

J = imnoise(I, type, parameters) дозволяє задати додатково параметри шуму.

J = imnoise(I, 'gaussian', m, v) додає гаусівський білий шум з середнім значенням m і відхиленням v (за замовчуванням m = 0 і v = 0.001). Цей вид шуму видимий як на темних так і на світлих областях зображення у вигляді характерних точок.

J = imnoise(I, 'salt & pepper', d) дозволяє задати густину шуму 'сіль і перець' d (за замовчуванням 0.05) для включених і виключених пікселів. Цей вид шуму також може проявлятися як на світлих так і на темних ділянках зображення.

J = imnoise(I, 'speckle', v) додає мультиплікативну компоненту шуму, так що $J = I + nI$, де n рівномірно розподілений шум з середнім значенням 0 і дисперсією v (за замовчуванням 0.04). Цей шум невидний на темних ділянках зображення, але проявляється на його світлих ділянках.

1.3. Функції MatLab для фільтрації зображень

Медіанна фільтрація

Ефективним засобом фільтрації шуму типу 'сіль і перець' є медіанна фільтрація. Функція $B = \text{medfilt2}(A, [m,n])$ фільтрує матрицю зображення A використовуючи маску фільтра розміром $m \times n$. Піксели початкового зображення, які попадають у вікно фільтра сортуються в порядку зростання. Центральному пікселю присвоюють значення пікселя який знаходиться посередині вектора. Маску застосовують нерекурсивно до всього зображення.

Чим більший розмір вікна, тим краще придушується шум, але при цьому зображення розмивається.

Лінійна фільтрація

Функція

$h = \text{fspecial}(\text{type})$

створює маску двовимірного фільтра заданого типу type у вигляді матриці h , яка зберігає коефіцієнти фільтра. Можливі наступні типи фільтрів (масок): 'average', 'gaussian', 'laplacian', 'log', 'sobel', 'prewitt', 'unsharp'.

Фільтр типу 'average' відноситься до класу фільтрів низьких частот. Він може використовуватися для зменшення впливу на зображення, проте його застосування приводить до розмиття зображення.

Фільтр Гауса 'gaussian' також є фільтром НЧ, але він менше розмиває зображення.

Фільтр Лапласа 'laplacian' це фільтр високих частот. Він може використовуватися для виділення границь (перепадів) зображень по всім напрямкам.

Фільтр 'log' – фільтр ВЧ здатний виділяти більш різкі переходи зображення ніж фільтр Лапласа.

Фільтр Собеля 'sobel' – служить для виділення горизонтальних границь об'єктів зображення з допомогою маски

$$h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Для виділення вертикальних границь достатньо транспонувати маску h .

Фільтр Превіта 'prewitt' – служить для виділення горизонтальних границь об'єктів зображення з допомогою маски

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}.$$

Для виділення вертикальних границь достатньо транспонувати маску h .

Фільтр підвищення різкості 'unsharp' має такі коефіцієнти

$$h = \frac{1}{a+1} \cdot \begin{bmatrix} -a & a-1 & -a \\ a-1 & a+5 & a-1 \\ -a & a-1 & -a \end{bmatrix}, \text{ де параметр } a \text{ вибирається в межах від } 0 \text{ до } 1.$$

Для здійснення фільтрації використовується функція imfilter :

$J = \text{imfilter}(I, h)$,

де I – вхідне зображення, h – маска фільтра, J – відфільтроване зображення.

Формат bmp-файлу

Отримані від CCD- або CMOS-сенсора зображення після обробки повинні зберігатися у Flash-пам'яті пристрою або на змінній карті пам'яті. Щоб ці дані можна було прочитати на іншому пристрої (наприклад, ПК) первинне зображення (Raw Image) повинне бути записане в певному графічному форматі. Розглянемо найпростіший формат графічних файлів.

Формат bmp (від англ. BitMaP – бітова карта або бітовий масив) представляє нестиснуте (в основному) зображення. Цей файл складається з чотирьох частин: заголовку, інформаційного заголовку, таблиці кольорів (палітри) і даних зображення. Якщо у файлі зберігається зображення з глибиною кольору 24 біта (truecolor), то таблиця кольорів може бути відсутня.

Структура BMP-файлу

Ім'я	Довжина, байт	Зміщення, байт	Опис
Заголовок файлу (BitmapFileHeader)			
Type	2	0	Тип файлу. Повинний бути 'BM' (0x424D).
Size	4	2	Розмір файлу у байтах.
Reserved 1	2	6	Зарезервовано. Повинні бути нулями.
Reserved 2	2	8	Зарезервовано. Повинні бути нулями.
OffsetBits	4	10	Зміщення зображення від початку файлу в байтах. Показує де починається сам бітовий масив відносно початку файлу.
Інформаційний заголовок (BitmapInfoHeader)			
Size	4	14	Довжина інформаційного заголовку в байтах (переважно рівна 0x00000028 – 40 байт).
Width	4	18	Ширина зображення в пікселях.
Height	4	22	Висота зображення в пікселях.
Planes	2	26	Число площин. Завжди рівне 1.
BitCount	2	28	Глибина кольору в бітах на піксел.
Compression	4	30	Тип компресії (переважно рівне 0 – немає компресії).
SizeImage	4	34	Розмір масиву зображення в байтах.
XpelsPerMeter	4	38	Горизонтальна роздільна здатність в пікселях на метр. Важливі для збереження масштабу від сканованих зображень. Зображення створені з допомогою графічних редакторів, переважно, мають в цих полях нулі.
YpelsPerMeter	4	42	Вертикальна роздільна здатність в пікселях на метр. Важливі для збереження масштабу від сканованих зображень. Зображення створені з допомогою графічних редакторів, переважно, мають в цих полях нулі.
ColorsUsed	4	46	Число використаних кольорів (переважно рівне 0 – для 24-бітних зображень, 256 – для 8 бітних).
ColorsImportant	4	50	Число основних кольорів (переважно рівне 0 – для 24-бітних зображень, 256 – для 8 бітних).
Таблиця кольорів (палітра) (ColorTable)			
ColorTable	1024	54	256 елементів по 4 байти (може бути відсутньою, для 8-бітних зображень має бути присутньою).
Дані зображення (Bitmap Array)			
Image	Size	1078	Зображення, записане по рядкам зліва направо і знизу вверх у форматі RGB (1-й байт B, 2-й байт G, 3-й байт R) – для 24 бітних зображень. Для 8-бітних зображень - по рядкам зліва направо і знизу вверх у форматі 1 байт. Довжина кожного рядка вирівняна на границі 4 байт, тобто при довжині рядка некратній чотирьом, вона доповнюється нулями до кратності.

Для 8-бітних зображень таблиця кольорів складається з 256 елементів, по 4 байти кожен. Елементи мають наступний формат:

1-й байт	2-й байт	3-й байт	4-й байт
ii	ii	ii	255

де $ii = 0 - 255$.

Для запису/читання файлів у MatLab використовуються функції `fwrite` і `fread` відповідно. Перш ніж здійснювати запис потрібно відкрити файл з допомогою функції `fopen`:

`fid = fopen(filename, permission)`

тут `filename` – ім'я файлу, `permission` – режим відкриття ('r' – читання, 'w' – запис, 'b' – бінарний файл, 't' – текстовий). Функція повертає ідентифікатор файлу `fid`.

Щоб записати дані у файл можна використати функцію `fwrite`:

`count = fwrite(fid, data, precision)`

тут `fid` – ідентифікатор файлу, в який здійснюється запис; `data` – дані, які потрібно записати (може бути матриця або скаляр); `precision` – тип даних у байтах ('uchar' - ціле число розміром 1 байт, 'int16' – ціле число розміром 2 байти, 'int32' – ціле число розміром 4 байти, 'float32' – число з плаваючою комою розміром 4 байти, 'double' – число з плаваючою комою розміром 8 байт). Функція повертає значення `count` – кількість успішно записаних елементів.

Після завершення роботи з файлом його потрібно закрити з допомогою функції `fclose`:

`st = fclose(fid)`

тут `fid` – ідентифікатор файлу. Функція повертає значення `st` – результат закриття файлу (0 – успішне, -1 – неуспішне).

Приклад програми формування bmp-файлу

```
I = imread('E:\xxxxxx.tiff');
```

```
% Визначити висоту і ширину зображення
```

```
[m, n] = size(I);
```

```
H_I = m; % Висота
```

```
W_I = n; % Ширина
```

```
% Доповнити кількість байт в рядку до кратних 4
```

```
W_Real = W_I + (W_I - fix(W_I/4)*4); % Справжній розмір рядка в байтах
```

```
% Обчислити розмір файлу
```

```
File_Size = 54 + W_Real * H_I + 256*4;
```

```
% Обчислити розмір зображення в байтах
```

```
Image_Size = W_Real * H_I;
```

```
% Зміщення зображення від початку файлу
```

```
Offset = 54 + 256*4;
```

```
% Відкрити файл для запису
```

```
d = fopen('my.bmp', 'wb');
```

```
%////////////////////////////////////
```

```
% Заголовок файлу (BitmapFileHeader)
```

```
% Заповнити поле Type
```

```
fwrite(d, 'B', 'uchar');
```

```
fwrite(d, 'M', 'uchar');
```

```
% Заповнити поле Size
```

```
fwrite(d, File_Size, 'int32');
```

```
% Заповнити поле Reserved 1
```

```
fwrite(d, 0, 'int16');
```

```

% Заповнити поле Reserved 2
fwrite(d, 0, 'int16');

% Заповнити поле OffsetBits
fwrite(d, Offset, 'int32');
%////////////////////////////////////
% Інформаційний заголовок (BitMapInfoHeader)
% Заповнити поле Size
fwrite(d, 40, 'int32');

% Заповнити поле Width
fwrite(d, W_I, 'int32');

% Заповнити поле Height
fwrite(d, H_I, 'int32');

% Заповнити поле Planes
fwrite(d, 1, 'int16');

% Заповнити поле BitCount
fwrite(d, 8, 'int16');

% Заповнити поле Compression
fwrite(d, 0, 'int32');

% Заповнити поле SizeImage
fwrite(d, Image_Size, 'int32');

% Заповнити поле XpelsPerMeter
fwrite(d, 0, 'int32');

% Заповнити поле YpelsPerMeter
fwrite(d, 0, 'int32');

% Заповнити поле ColorsUsed
fwrite(d, 256, 'int32');

% Заповнити поле ColorsImportant
fwrite(d, 256, 'int32');
%////////////////////////////////////
% Таблиця кольорів (палітра) (ColorTable)
for ii = 0 : 255
    Temp = [ii ii ii 255];
    fwrite(d, Temp, 'uchar');
end;
%////////////////////////////////////
% Дані зображення (BitMap Array)
Temp = zeros(1, W_Real);

for ii = H_I : -1 : 1
    Temp(1:W_I) = I(ii, 1:W_I);
    fwrite(d, Temp, 'uchar');
end;
%////////////////////////////////////
fclose(d);

```


3. ЗАВДАННЯ

1. Ознайомитись з теоретичним матеріалом.
2. Завантажити файл вказаний в завданні відповідно до свого варіанту.
3. Вивести основну інформацію про графічний файл.
4. Накласти на зображення шуми gaussian, salt & pepper, speckle.
5. Провести лінійну фільтрацію зашумленого зображення фільтрами 'average', 'gaussian', 'sobel', 'prewitt', 'unsharp'. Зробити висновки про ефективність фільтрів.
6. Провести медіанну фільтрацію. Зробити висновки про ефективність фільтру.
7. Написати програму збереження відфільтрованого зображення у bmp-файлі.
8. Переконатися у працездатності програми.

Табл.

Варіант	Назва файлу	Варіант	Назва файлу
1	cameraman1.tiff	11	cameraman1.tiff
2	wbarb.tiff	12	wmandril.tiff
3	chess.tiff	13	belmont2.tiff
4	belmont1.tiff	14	wbarb.tiff
5	board1.tiff	15	chess.tiff
6	detfingr.tiff	16	flowers1.tiff
7	moon.tiff	17	belmont1.tiff
8	belmont2.tiff	18	moon.tiff
9	wmandril.tiff	19	detfingr.tiff
10	flowers1.tiff	20	board1.tiff

4. ЗМІСТ ЗВІТУ

1. Мета роботи.
2. Повний текст завдання.
3. Лістинг програми.
4. Графік оригінального зображення.
5. Графік відфільтрованого зображення.
6. Висновок.

5. КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які типи зображень Ви знаєте ?
2. Як здійснюється лінійна фільтрація зображень ?
3. Які типи лінійних фільтрів Ви знаєте. Яке їх призначення ?
4. Які типи нелінійних фільтрів Ви знаєте ?
5. Як здійснюється медіанна та рангова фільтрація

6. СПИСОК ЛІТЕРАТУРИ

1. Шрюфер Э. Обработка сигналов: цифровая обработка дискретизированных сигналов. – К.: Либідь, 1995. – 320 с.
2. Сергиенко А. Б. Цифровая обработка сигналов. – СПб.: Питер, 2002. – 608 с.
3. Лозинський А. О., Мороз В. І., Паранчук Я. С. Розв'язування задач електромеханіки в середовищах пакетів MathCAD і MATLAB: Навчальний посібник. – Львів: Видавництво Національного університету "Львівська політехніка", 2000–166 с.
4. Лазарев Ю. Ф. MatLab 5.x. – К.: BHV, 2000. – 384 с.

НАВЧАЛЬНЕ ВИДАННЯ

**ДОСЛІДЖЕННЯ МЕТОДІВ ОБРОБКИ
ЗОБРАЖЕНЬ В MATLAB**

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторної роботи № 3
з курсу “ Системи запису та відтворення інформації ”
для студентів базового напрямку 6.170102
“ Системи технічного захисту інформації ”

Укладачі: Совин Ярослав Романович, к.т.н., доцент

Редактор

Комп'ютерне видання

Підписано до друку
Формат 70 x 100 ¹/₁₆. Папір офсетний.
Друк на різнографі. Умовн. друк. арк. Обл.-вид. Арк.
Наклад. Прим. Зам.

Поліграфічний центр
Видавництва Національного університету "Львівська політехніка"
Вул. Ф.Колеси, 2, 79000, Львів