

## Приклад програми для візуалізації і перетворення форматів відеозображень

```
% ЛАБОРАТОРНА РОБОТА №1
% ДОСЛІДЖЕННЯ МЕТОДІВ ВВЕДЕННЯ ВІДЕОЗБРАЖЕНЬ В КОМП'ЮТЕР % ТА ЇХ ЗБЕРІГАННЯ В
ЦИФРОВІЙ ФОРМІ

% ЗАВАНТАЖЕННЯ ПОЧАТКОВОГО ВІДЕОЗБРАЖЕННЯ
RGB1=imread('apple.bmp');

% ПЕРЕТВОРЕННЯ ТИПУ ВІДЕОЗБРАЖЕННЯ
I1=rgb2gray(RGB1);
imwrite(I1,'apple_gray.jpg');
BW1=im2bw(RGB1, 0.5);
imwrite(BW1,'apple_bw.tif');

% ВИВЕДЕННЯ РЕЗУЛЬТАТІВ
subplot(2,2,1), imshow(RGB1);
title('ПОЧАТКОВЕ ВІДЕОЗБРАЖЕННЯ');
subplot(2,2,2), imshow(I1);
title('НАПІВТОНОВЕ ВІДЕОЗБРАЖЕННЯ');
subplot(2,2,3), imshow(BW1);
title('ДВОГРАДАЦІЙНЕ ВІДЕОЗБРАЖЕННЯ');
```

## Результати роботи програми

ПОЧАТКОВЕ ВІДЕОЗБРАЖЕННЯ



НАПІВТОНОВЕ ВІДЕОЗБРАЖЕННЯ



ДВОГРАДАЦІЙНЕ ВІДЕОЗБРАЖЕННЯ



## Приклад програми для стиснення цифрових відеозображень за JPEG-алгоритмом

```

% ЛАБОРАТОРНА РОБОТА №2
% СТИСНЕННЯ ЦИФРОВИХ ВІДЕОЗОБРАЖЕНЬ ЗА JPEG-АЛГОРИТМОМ
%
% ВВЕДЕННЯ ПОЧАТКОВИХ ДАНИХ
prompt={'ІМ"Я ФАЙЛА ЗОБРАЖЕННЯ',...
'РОЗМІР ЗОБРАЖЕННЯ КxК ДИСКР. ТОЧОК: K=32,64,128,256,512',...
'РОЗМІР БЛОКУ ДКП NxN ДИСКР. ТОЧОК: N=4,8,16,32,64,128,256,512',...
'МЕТОД ОБЧИСЛЕННЯ ДКП: 1-СУМА; 2-МАТРИЦІ',...
'КІЛЬКІСТЬ НЕНУЛЬОВИХ КОЕФ. ДКП: M=1,2,...,N; 0-МЕТОД НЕ ВИКОРИСТ.',...
'ПОРОГ ДЛЯ НЕНУЛЬОВИХ КОЕФ. ДКП: P=1,...,255; 0-МЕТОД НЕ ВИКОРИСТ.',...
'ВИВЕДЕННЯ РЕЗУЛЬТАТІВ: 1-ЗОБРАЖЕННЯ; 2-ТАБЛИЦЯ; 3-ЗОБР. І ТАБЛ.'};
def={'apple.bmp', '256', '8', '1', '0', '0', '3'};
dlgTitle='ЛАБОРАТОРНА РОБОТА №2';
lineNo=1;
AddOpts.Resize='on';
AddOpts.WindowStyle='normal';
AddOpts.Interpreter='tex';
Answer=inputdlg(prompt,dlgTitle,lineNo,def,AddOpts);
ImageName=Answer{1}; % ІМ"Я ФАЙЛА ЗОБРАЖЕННЯ
K=str2num(Answer{2}); % РОЗМІР ЗОБРАЖЕННЯ КxК ДИСКР. ТОЧОК
N=str2num(Answer{3}); % РОЗМІР БЛОКУ ДКП NxN ДИСКР. ТОЧОК
TypeCalc=str2num(Answer{4}); % МЕТОД ОБЧИСЛЕННЯ ДКП
M=str2num(Answer{5}); % КІЛЬКІСТЬ НЕНУЛЬОВИХ КОЕФ. ДКП
P=str2num(Answer{6}); % ПОРОГ ДЛЯ НЕНУЛЬОВИХ КОЕФ. ДКП
TypeOutput=str2num(Answer{7}); % ВИВЕДЕННЯ РЕЗУЛЬТАТІВ

% ІНІЦІАЛІЗАЦІЯ ЗМІННИХ
OrigImage=zeros(K,K);
RestoreImage=zeros(K,K);
CoefDCT=zeros(K,K);
CoefDCTCompress=zeros(K,K);
CoefMul=zeros(N,N);

% ЗАВАНТАЖЕННЯ ПОЧАТКОВОГО ЗОБРАЖЕННЯ
RGB=imread(ImageName);
II=rgb2gray(RGB);
OrigImage=II(1:K,1:K);

% ОБЧИСЛЕННЯ ДКП
switch TypeCalc
case 1
    Time1=cputime;
    fun=@dct2;
    CoefDCT=blkproc(OrigImage, [N N], fun);
    Time2=cputime;
    fprintf('\nЧАС ОБЧИСЛЕННЯ ПРЯМОГО ДКП (СУМА) %7.3f СЕКУНД\n',Time2-Time1);
case 2
    Time1=cputime;
    CoefMul=dctmtx(N);
    fun = inline('P1*double(x)*ctranspose(P1)', 1);
    CoefDCT=blkproc(OrigImage, [N N], fun, CoefMul);
    Time2=cputime;
    fprintf('\nЧАС ОБЧИСЛЕННЯ ПРЯМОГО ДКП (МАТРИЦІ) %7.3f СЕКУНД\n',Time2-Time1);
end

% СТИСНЕННЯ ЗОБРАЖЕННЯ
if P~=0
    CoefDCTCompress=CoefDCT;
    CoefDCTCompress((abs(CoefDCTCompress))<P)=0;
end
if M~=0
    h = waitbar(0,'СТИСНЕННЯ ЗОБРАЖЕННЯ ...');
    CoefDCTCompress=zeros(K,K);
    for i=1:(K/N)

```

```

        for j=1:(K/N)
            waitbar((i-1)*(K/N)+j)/((K/N)*(K/N)),h);
            CoefDCTCompress((i-1)*N+1):(i-1)*N+M,
((j-1)*N+1):(j-1)*N+M)=...
            CoefDCT(((i-1)*N+1):(i-1)*N+M),
((j-1)*N+1):(j-1)*N+M));
        end
    end
    close(h);
end
if (P==0)&(M==0)
    CoefDCTCompress=CoefDCT;
end

% ВІДНОВЛЕННЯ ЗОБРАЖЕННЯ
switch TypeCalc
case 1
    Time1=cputime;
    fun=@idct2;
    RestoreImage=uint8(blkproc(CoefDCTCompress,[N N],fun));
    % RestoreImage=uint8(idct2(CoefDCTCompress));
    Time2=cputime;
    fprintf('ЧАС ОБЧИСЛЕННЯ ОБЕРНЕНОГО ДКП (СУМА) %7.3f СЕКУНД\n',Time2-Time1);
case 2
    Time1=cputime;
    fun = inline('uint8(ctranspose(P1)*x*P1)', 1);
    RestoreImage=blkproc(CoefDCTCompress,[N N],fun,CoefMul);
    Time2=cputime;
    fprintf('ЧАС ОБЧИСЛЕННЯ ОБЕРНЕНОГО ДКП (МАТРИЦІ) %7.3f СЕКУНД\n',Time2-Time1);
end

% ОБЧИСЛЕННЯ ПОХИБКИ, ЩО ВИНИКЛА В РЕЗУЛЬТАТІ СТИСНЕННЯ
ErrorDCTArray=abs(double(RestoreImage)-double(OrigImage));
ErrorDCTMean=mean2(ErrorDCTArray);
ErrorDCTSKO=sqrt((sum(sum(ErrorDCTArray.^2)))/(K*K));
ErrorDCTMax=max(max(ErrorDCTArray));
ErrorDCTMin=min(min(ErrorDCTArray));

% ОБЧИСЛЕННЯ КОЕФІЦІЕНТУ СТИСНЕННЯ
RCompress=prod(size(CoefDCTCompress))/nnz(CoefDCTCompress);

% ВИВЕДЕННЯ РЕЗУЛЬТАТІВ
if ((TypeOutput==2)|(TypeOutput==3))
    fprintf('ДИНАМІЧНИЙ ДІАПАЗОН ЯСКРАВОСТІ 255 ДИСКРЕТНИХ РІВНЕЙ\n');
    fprintf('ПОХИБКА, ЩО ВИНИКЛА В РЕЗУЛЬТАТІ СТИСНЕННЯ\n');
    fprintf('СЕРЕДНЄ ЗНАЧЕННЯ %7.3f ДИСКРЕТНИХ РІВНЕЙ\n', ErrorDCTMean);
    fprintf('СЕРЕДНЬОКВАДРАТИЧНЕ ЗНАЧЕННЯ %7.3f ДИСКРЕТНИХ РІВНЕЙ\n', ErrorDCTSKO);

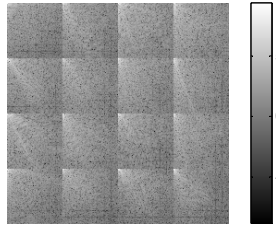
    fprintf('МІНІМАЛЬНЕ ЗНАЧЕННЯ %7.3f ДИСКРЕТНИХ РІВНЕЙ\n', ErrorDCTMin);
    fprintf('МАКСИМАЛЬНЕ ЗНАЧЕННЯ %7.3f ДИСКРЕТНИХ РІВНЕЙ\n', ErrorDCTMax);
    fprintf('СТИСНЕННЯ ЗОБРАЖЕННЯ (КІЛЬКІСТЬ ВСІХ КОЕФІЦІЕНТІВ ДКП/КІЛЬКІСТЬ НУЛЬОВИХ
КОЕФІЦІЕНТІВ)\n');
    fprintf('%7.3f РАЗІВ\n', RCompress);
end
if ((TypeOutput==1)|(TypeOutput==3))
    subplot(2,2,1); imshow(OrigImage);
    subplot(2,2,2); imshow(log(abs(CoefDCT)),[]);
    colormap(gray(256)); colorbar;
    subplot(2,2,3); imshow(RestoreImage);
    subplot(2,2,4); imshow(log(abs(CoefDCTCompress)),[]);
    colormap(gray(256)); colorbar;
end
end

```

## Результати роботи програми



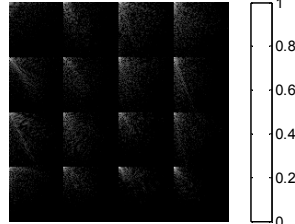
Початкове відеозображення



Частотні коефіцієнти спектра,  
отримані в результаті ДКП



Відеозображення, відновлене  
після стиснення



Частотні коефіцієнти спектра  
після стиснення

ЧАС ОБЧИСЛЕННЯ ПРЯМОГО ДКП (СУМА) 0.440 СЕКУНД  
ЧАС ОБЧИСЛЕННЯ ОВЕРНЕНОГО ДКП (СУМА) 0.490 СЕКУНД  
ДИНАМІЧНИЙ ДІАПАЗОН ЯСКРАВСТІ 255 ДИСКРЕТНИХ РІВНЕЙ  
ПОХИБКА, ЩО ВИНИКЛА В РЕЗУЛЬТАТІ СТИСНЕННЯ  
СЕРЕДНЄ ЗНАЧЕННЯ 2.815 ДИСКРЕТНИХ РІВНЕЙ  
СЕРЕДНЬОКВАДРАТИЧНЕ ЗНАЧЕННЯ 3.708 ДИСКРЕТНИХ РІВНЕЙ  
МІНІМАЛЬНЕ ЗНАЧЕННЯ 0.000 ДИСКРЕТНИХ РІВНЕЙ  
МАКСИМАЛЬНЕ ЗНАЧЕННЯ 27.000 ДИСКРЕТНИХ РІВНЕЙ  
СТИСНЕННЯ ЗОБРАЖЕННЯ (КІЛЬКІСТЬ ВСІХ КОЕФІЦІЄНТІВ ДКП/КІЛЬКІСТЬ НУЛЬОВИХ КОЕФІЦІЄНТІВ)  
4.248 РАЗІВ

## Приклад програми для дослідження методів фільтрації шумів на цифрових відеозображеннях в інтелектуальних системах

```

% ЛАБОРАТОРНА РОБОТА №3
% ДОСЛІДЖЕННЯ МЕТОДІВ ФІЛЬТРАЦІЇ ШУМІВ НА ЦИФРОВИХ
% ВІДЕОЗОБРАЖЕННЯХ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ

ImageName='saturn.tif';           % ІМ'Я ФАЙЛА ЗОБРАЖЕННЯ

% ВИЗНАЧЕННЯ ШУМУ
Tsh='gaussian';                  % ТИП ШУМУ НА ЗОБРАЖЕННІ
% 'gaussian' - "БІЛИЙ" ШУМ З НОРМАЛЬНИМ РОЗПОДІЛОМ
M=0;                             % СЕРЕДНЄ ЗНАЧЕННЯ ШУМУ
V=0.03;                           % ДИСПЕРСІЯ ШУМУ

% Tsh='salt & pepper';          % ТИП ШУМУ НА ЗОБРАЖЕННІ
% 'salt & pepper' - ШУМ У ВИГЛЯДІ БІЛИХ І ЧОРНИХ ТОЧОК
% D=0.05;                        % ЩІЛЬНІСТЬ ШУМУ НА ЗОБРАЖЕННІ

% Tsh='speckle';                % ТИП ШУМУ НА ЗОБРАЖЕННІ
% 'speckle' - МУЛЬТИПЛІКАТИВНИЙ ШУМ
% V=0.01;                        % ДИСПЕРСІЯ ШУМУ

% СТВОРЕННЯ ФІЛЬТРА
Tfilter='average';              % ТИП ФІЛЬТРА
% 'average' - УСЕРЕДНЮЮЧИЙ ФІЛЬТР
Hsize=3;                        % РОЗМІР КВАДРАТНОЇ МАСКИ ФІЛЬТРА
Filter=fspecial(Tfilter,Hsize); % СТВОРЕННЯ МАСКИ ФІЛЬТРА

% Tfilter='gaussian';           % ТИП ФІЛЬТРА
% 'gaussian' - ГАУСОВ ФІЛЬТР НИЖНІХ ЧАСТОТ
% РАДІУС МАСКИ ФІЛЬТРА, РОЗМІР КВАДРАТНОЇ МАСКИ Radius*2+1
% Radius=5;
% СТВОРЕННЯ МАСКИ ФІЛЬТРА
% Filter=fspecial(Tfilter,Radius);

% ЗАВАНТАЖЕННЯ ПОЧАТКОВОГО ЗОБРАЖЕННЯ
OrigImage=imread(ImageName);
if isrgb(OrigImage)
    OrigImage=rgb2gray(OrigImage);
end

% ДОДАВАННЯ ШУМУ ДО ЗОБРАЖЕННЯ
NoiseImage = imnoise(OrigImage,Tsh,M,V);
% NoiseImage = imnoise(OrigImage,Tsh,D);
% NoiseImage = imnoise(OrigImage,Tsh,V);

% ФІЛЬТРАЦІЯ ЗОБРАЖЕННЯ
% УСЕРЕДНЮЮЧИЙ ФІЛЬТР З КВАДРАТНОЮ МАСКОЮ
% Hsize x Hsize ТОЧОК
RestoreImage=imfilter(NoiseImage, Filter);
% МЕДІАННИЙ ФІЛЬТР
% РОЗМІР МАТРИЦІ ДЛЯ ВИЗНАЧЕННЯ СУСІДНИХ ТОЧОК
% Mfilter=[3 3];
% RestoreImage=medfilt2(NoiseImage,Mfilter);

% ОБЧИСЛЕННЯ ВІДМІННОСТЕЙ МІЖ ВІДНОВЛЕНИМ І
% ПОЧАТКОВИМ ЗОБРАЖЕННЯМ
ErrorImage=uint8(abs(double(RestoreImage)-double(OrigImage)));
DeltaMean=double(mean2(ErrorImage));
DeltaMax=double(max(max(ErrorImage)));

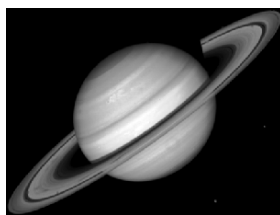
% ВИВЕДЕННЯ РЕЗУЛЬТАТІВ
fprintf('ДОСЛІДЖЕННЯ МЕТОДІВ ФІЛЬТРАЦІЇ ВІДЕОІНФОРМАЦІЇ В ІНТЕЛЕКТУАЛЬНИХ
СИСТЕМАХ\n');
fprintf('МАКСИМАЛЬНЕ ЗНАЧЕННЯ ПОХИБКА ВІДНОВЛЕННЯ %7.3f ДИСКРЕТНИХ РІВНЕЙ\n',
DeltaMax);
fprintf('СЕРЕДНЄ ЗНАЧЕННЯ ПОХИБКА ВІДНОВЛЕННЯ %7.3f ДИСКРЕТНИХ РІВНЕЙ\n', DeltaMean);
subplot(2,2,1); imshow(OrigImage);

```

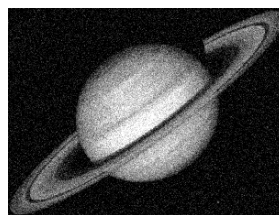
```
title('ПОЧАТКОВЕ ЗОБРАЖЕННЯ');  
subplot(2,2,2); imshow(NoiseImage);  
title('ЗОБРАЖЕННЯ З ШУМОМ');  
subplot(2,2,3); imshow(RestoreImage);  
title('ВІДНОВЛЕНЕ ЗОБРАЖЕННЯ');  
subplot(2,2,4); imshow(ErrorImage);  
title('ПОХИБКА ВІДНОВЛЕННЯ');
```

Результати роботи програми  
(фільтрація “білого” шуму з нормальним розподілом та шуму у вигляді чорних і білих точок)

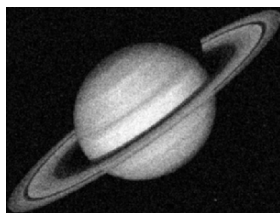
ПОЧАТКОВЕ ЗОБРАЖЕННЯ



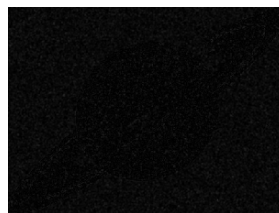
ЗОБРАЖЕННЯ З ШУМОМ



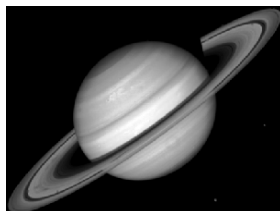
ВІДНОВЛЕНЕ ЗОБРАЖЕННЯ



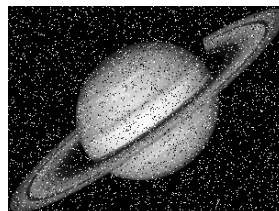
ПОХИБКА ВІДНОВЛЕННЯ



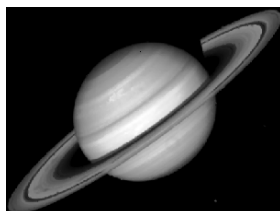
ПОЧАТКОВЕ ЗОБРАЖЕННЯ



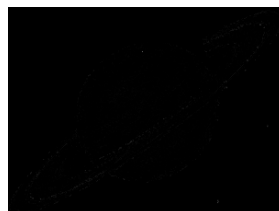
ЗОБРАЖЕННЯ З ШУМОМ



ВІДНОВЛЕНЕ ЗОБРАЖЕННЯ



ПОХИБКА ВІДНОВЛЕННЯ



**Приклад програми для дослідження методів виділення  
контурів об'єктів на цифрових відеозображеннях в  
інтелектуальних системах**

```

% ЛАБОРАТОРНА РОБОТА №4
% МЕТОДИ ВИДІЛЕННЯ КОНТУРІВ ОБ'ЄКТІВ НА ЦИФРОВИХ
% ВІДЕОЗОБРАЖЕННЯХ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ
ImageName='saturn.tif';           % ІМ'Я ФАЙЛА ЗОБРАЖЕННЯ

% ВИБІР МЕТОДУ ВІДІЛЕННЯ КОНТУРІВ
Type='sobel';           % МЕТОД СОБЕЛЯ
                        % ПАРАМЕТРИ МЕТОДУ
Thresh=0.05;           % ПОРІГ ДЛЯ ВИЗНАЧЕННЯ КОНТУРА
Direction='both';     % НАПРЯМОК РОЗТАШУВАННЯ КОНТУРІВ
                        % 'horizontal' - ГОРИЗОНТАЛЬНИЙ
                        % 'vertical' - ВЕРТИКАЛЬНИЙ
                        % 'both' - В ОБОХ НАПРЯМКАХ

% Type='prewitt';     % МЕТОД ПРЕВІТА
%                     % ПАРАМЕТРИ МЕТОДУ
% Thresh=0.05;       % ПОРІГ ДЛЯ ВИЗНАЧЕННЯ КОНТУРА
% Direction='both';  % НАПРЯМОК РОЗТАШУВАННЯ КОНТУРІВ
%                     % 'horizontal' - ГОРИЗОНТАЛЬНИЙ
%                     % 'vertical' - ВЕРТИКАЛЬНИЙ
%                     % 'both' - В ОБОХ НАПРЯМКАХ

% Type='roberts';    % МЕТОД РОБЕРТСА
%                   % ПАРАМЕТРИ МЕТОДУ
% Thresh=0.05;      % ПОРІГ ДЛЯ ВИЗНАЧЕННЯ КОНТУРА
% Direction='both'; % НАПРЯМОК РОЗТАШУВАННЯ КОНТУРІВ
%                   % 'horizontal' - ГОРИЗОНТАЛЬНИЙ
%                   % 'vertical' - ВЕРТИКАЛЬНИЙ
%                   % 'both' - В ОБОХ НАПРЯМКАХ

% Type='log';        % МЕТОД ЛАПЛАСИАНА КРИВОЇ ГАУССА
%                   % ПАРАМЕТРИ МЕТОДУ
% Thresh=0.005;     % ПОРІГ ДЛЯ ВИЗНАЧЕННЯ КОНТУРА
% Direction='both'; % НАПРЯМОК РОЗТАШУВАННЯ КОНТУРІВ
%                   % 'horizontal' - ГОРИЗОНТАЛЬНИЙ
%                   % 'vertical' - ВЕРТИКАЛЬНИЙ
%                   % 'both' - В ОБОХ НАПРЯМКАХ

% Type='canny';     % МЕТОД КАННИ
%                   % ПАРАМЕТРИ МЕТОДУ
% Thresh=0.05;      % ПОРІГ ДЛЯ ВИЗНАЧЕННЯ КОНТУРА
% Direction='both'; % НАПРЯМОК РОЗТАШУВАННЯ КОНТУРІВ
%                   % 'horizontal' - ГОРИЗОНТАЛЬНИЙ
%                   % 'vertical' - ВЕРТИКАЛЬНИЙ
%                   % 'both' - В ОБОХ НАПРЯМКАХ

% ЗАВАНТАЖЕННЯ ПОЧАТКОВОГО ЗОБРАЖЕННЯ
OrigImage=imread(ImageName);
if isrgb(OrigImage)
    OrigImage=rgb2gray(OrigImage);
end

% ДОДАВАННЯ ШУМУ ДО ЗОБРАЖЕННЯ
NoiseImage = imnoise(OrigImage,'gaussian',0,0.01);

% ФІЛЬТРАЦІЯ ЗОБРАЖЕННЯ
% УСЕРЕДНЮЮЧИЙ ФІЛЬТР З КВАДРАТНОЮ МАСКОЮ Hsize x Hsize ТОЧОК
Tfilter='average'; % ТИП ФІЛЬТРА
Hsize=5;           % РОЗМІР КВАДРАТНОЇ МАСКИ ФІЛЬТРА
Filter=fspecial(Tfilter,Hsize); % СТВОРЕННЯ МАСКИ ФІЛЬТРА
RestoreImage=imfilter(NoiseImage, Filter);

% ВИДІЛЕННЯ КОНТУРІВ
% ВИКОРИСТАННЯ ЗАДАНИХ ПАРАМЕТРІВ
% BW1=edge(OrigImage,Type,Thresh,Direction);

```



```

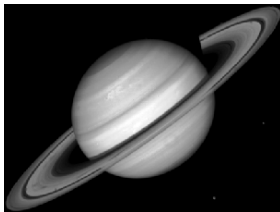
% BW2=edge (NoiseImage, Type, Thresh, Direction);
% АВТОМАТИЧНЕ ВИЗНАЧЕННЯ ПОРОГУ
[BW1, Tresh1]=edge (OrigImage, Type);
[BW2, Tresh2]=edge (NoiseImage, Type);
CountEdge1=nnz (double (BW1));
CountEdge2=nnz (double (BW2));

% ВИВЕДЕННЯ РЕЗУЛЬТАТІВ
fprintf('\nМЕТОДИ ВИДІЛЕННЯ КОНТУРІВ ОБ'ЄКТІВ НА ВІДЕОЗОБРАЖЕННЯХ\n');
fprintf('ЗАГАЛЬНА ДОВЖИНА КОНТУРІВ НА ПОЧАТКОВОМУ ЗОБРАЖЕННІ %7.0f ДИСКРЕТНИХ ТОЧОК\n', CountEdge1);
fprintf('ЗАГАЛЬНА ДОВЖИНА КОНТУРІВ НА ЗОБРАЖЕННІ З ШУМОМ %7.0f ДИСКРЕТНИХ ТОЧОК\n', CountEdge2);
fprintf('ПОРІГ ДЛЯ ВИДІЛЕННЯ КОНТУРІВ %7.3f %7.3f ДИСКРЕТНИХ РІВНІВ\n', Tresh1, Tresh2);
subplot(2,2,1); imshow(OrigImage);
title('ПОЧАТКОВЕ ЗОБРАЖЕННЯ');
subplot(2,2,2); imshow(NoiseImage);
title('ЗОБРАЖЕННЯ З ШУМОМ');
subplot(2,2,3); imshow(BW1);
title('КОНТУРИ НА ПОЧАТК. ЗОБР. ');
subplot(2,2,4); imshow(BW2);
title('КОНТУРИ НА ЗОБР. З ШУМОМ');

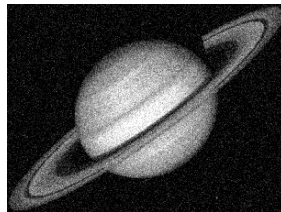
```

### Результати роботи програми

ПОЧАТКОВЕ ЗОБРАЖЕННЯ



ЗОБРАЖЕННЯ З ШУМОМ



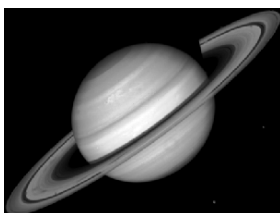
КОНТУРИ НА ПОЧАТК. ЗОБР.



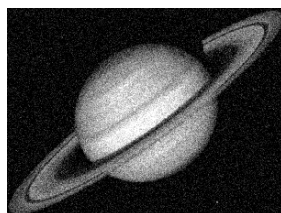
КОНТУРИ НА ЗОБР. З ШУМОМ



ПОЧАТКОВЕ ЗОБРАЖЕННЯ



ЗОБРАЖЕННЯ З ШУМОМ



КОНТУРИ НА ПОЧАТК. ЗОБР.



КОНТУРИ НА ЗОБР. ПІСЛЯ ФІЛЬТРАЦІЇ

