

## Zajęcia 5 – przetwarzanie obrazów (skrót i przykłady)

Ze strony [http://www.fis.agh.edu.pl/~gorczyca/dyd\\_pakiet\\_pliki.html](http://www.fis.agh.edu.pl/~gorczyca/dyd_pakiet_pliki.html) pobieramy pliki ze zdjęciami: Zdjecie1, ...Zdjecie8.

### Wczytywanie, wyświetlanie, zmiana typu.

Ogólna postać instrukcji wczytywania obrazu z pliku do macierzy:

Nazwa\_macierzy = **imread**(nazwa\_pliku); dla obrazu kolorowego typu indeksowanego funkcja ta wczytuje także drugą macierz - paletę (mapę) kolorów.

```
z1=imread('zdjecie1.jpg');
```

### Wyświetlanie obrazów

Do wyświetlenia obrazu służy instrukcja: **imshow** (macierz\_obrazu, macierz\_palety); a dla obrazów szarych wystarczy: **imshow** (macierz obrazu);

### Parametry obrazów cyfrowych

W Matlabie informacje o obrazie można uzyskać dzięki funkcji **imfinfo**:  
`imfinfo('zdjecie1.jpg');`

### Przykład 1

```
clear all; clc; close all;
% wczytanie grafiki z pliku jako obraz RGB
zdjecie1=imread('zdjecie1.jpg');
figure(1)
image(zdjecie1); %wyswietlenie obrazu RGB
input('konwersja rgb2ind ');
%konwersja z trybu RGB na tryb intensywnosc i mapa kolorow
[zdjecie2,cmap2] = rgb2ind(zdjecie1,256); % 256 - ilosc kolorow
zdjecie2d=double(zdjecie2);%zamiana na double
figure(2);
surf(zdjecie2d,'EdgeColor','none'); %
title('Polecenie surf');
input('teraz bedzie komenda view (0,90) ');
view(0,90)
input('dolozymy mape kolorow ');
colormap(cmap2);
figure(3);
imagesc(zdjecie2d); %
title('Polecenie image');
view(0,90)
colormap(cmap2);
axis off; % mozna wylaczyc osie
input('teraz zmniejszymy jasnoc 2 razy');
%konwersja z typu zmiennej int8 na double
zdjecie3d = double(zdjecie1);
zdjecie3c = zdjecie3d/2; %zmniejszenie jasnosci
zdjecie3=uint8(round(zdjecie3c));
figure(4)
imagesc(zdjecie3);
title('zdjecie przyciemnione');
input('zwiększymy jasnoc');
zdjecie3j = zdjecie3d*2; % zwiększenie jasnosci
zdjecie4=uint8(round(zdjecie3j));
figure(5)
imagesc(zdjecie4);
title('zdjecie rozjaśnione');
```

```

input('Zdjęcie w odcieniach szarości');
zdjecie5=rgb2gray(zdjecie1);
% h5=subplot(3,2,5);
h5=figure(7);
imagesc(zdjecie5);
title('odcienie szarości');
colormap(h5,gray); %spróbować to włączyć/wyłączyć
%można poeksperymentować zmieniając mapę kolorów np:
%colormap(jet)

```

## Przykład 2

```

%przykład przenikania 1 zdjęcia w drugie
clear all; close all; clc;
%wczytanie zdjec
zdjecie1=imread('zdjecie1.jpg');
zdjecie2=imread('zdjecie2.jpg');
%wyswietlenie zdjec
subplot(2,1,1);
%włączenie podwojnego buforowania
%Double buffering is the process of drawing into an offscreen
%pixel buffer and then blitting the buffer contents to
%the screen once the drawing is complete (instead of
%drawing directly to the screen, where the process of
%drawing is visible as it progresses).
set(gcf, 'Doublebuffer', 'on');
image(zdjecie1);
axis off;
axis image;
subplot(2,1,2);
image(zdjecie2);
axis off;
axis image;
%konwersja do typu double
zdjecie1d=double(zdjecie1);
zdjecie2d=double(zdjecie2);
%utworzenie 3-go plotu i mieszanie w petli 2 zdjec w roznych proporcjach
figure
for i=0:0.01:1
    zdjecie3d=i.*zdjecie1d+(1-i).*zdjecie2d;
    %konwersja do int8 zeby dalo sie wyswietlic procedura image
    zdjecie3=uint8(round(zdjecie3d));
    image(zdjecie3);
    axis off;
    axis image;
    pause(0.1);
end;

```

## Przykład 3

```

%owijanie kuli obrazem płaskim i obrót globusa
clear;
clc;
clf;
zdj1=imread('earth.png');
[zdj1i,cmap1] = rgb2ind(zdj1,256);
[x y z]=sphere;
surface(x,y,z,flipud(zdj1i), 'FaceColor','texturemap',...
    'EdgeColor','none',...
    'CDataMapping','direct');
colormap(cmap1);
axis off;
axis image;
view(0,0);
camva('manual');
for i=1:300

```

```

[x y z]= sph2cart(i/50, 22, 20);
campos([x y z]);
drawnow;
end

```

#### Przykład 4

```

%wsuwanie od góry
clear;
clc;
clf;
z1=imread('zdjecie1.jpg');
z2=imread('zdjecie2.jpg');
z3=z2;
s=size(z1,1);
for i=1:3:s(1)
    z3(1:i,:)=z1((s-i+1):s,:);
    image(z3);
    axis off;
    axis image;
    pause(0.01)
end;

```

#### Histogramy obrazu.

Wiele informacji o obrazie można uzyskać, rozpatrując jego **wybrane przekroje**. Obraz będący funkcją dwóch zmiennych jest trudny do interpretacji, chociaż bardzo plastycznie oddający charakter obrazu. W Matlabie obraz można przedstawić jako funkcję dwóch zmiennych lub uzyskać **przekroje** obrazów w poziomie, pionie lub względem dowolnej prostej. Linie, wzdłuż których mają być narysowane przekroje poziomów szarości w obrazie, mogą być zadane myszką lub przez podanie ich współrzędnych  $x_1, y_1$ .

#### %Przykład 5a - Obraz i jego histogram wzdłuż prostej poziomej

```

close all; clear all; clc;
[L1]=imread('komorki.tif');
figure; imshow(L1);
[mm,nn]=size(L1);
H=150;%polozenie linii
c=improfile(L1,[1 nn],[H H]);
line([1 nn],[H H],'Color',[0 1 0],'LineWidth',3);
set(gcf,'Color',[1 1 1]);
hold on;
plot(c(:, :, 1), 'k');
set(gcf,'Color',[1 1 1]);
xlabel('n','FontSize',15,'FontName','Arial CE');
ylabel('L(82,n)','FontSize',15,'FontName','Arial CE');

```

#### %Przykład 5b - Obraz i jego histogram wzdłuż prostej pionowej

```

[L1]=imread('komorki.tif');
figure; imshow(L1);
[mm,nn]=size(L1);
c=improfile(L1,[110 110],[1 mm]);
line([110 110],[1 mm],'Color',[1 1 1],'LineWidth',3);
set(gcf,'Color',[1 1 1]);
figure; plot(c(:, :, 1), 'k');
set(gcf,'Color',[1 1 1]);
xlabel('n','FontSize',15,'FontName','Arial CE');
ylabel('L(m,110)','FontSize',15,'FontName','Arial CE');

```

#### %Przykład 5c - Obraz i jego histogram wzdłuż prostej ukośnej

```

[L1]=imread('komorki.tif');

```

```

figure; imshow(L1);
[mm,nn]=size(L1);
c=improfile(L1,[80 240],[1 mm]);
line([80 240],[1 mm],'Color',[1 1 1],'LineWidth',3);
set(gcf,'Color',[1 1 1]);
figure; plot(c(:, :, 1), 'k');
set(gcf,'Color',[1 1 1]);
xlabel('n','FontSize',15,'FontName','Arial CE');
ylabel('L(m,n)','FontSize',15,'FontName','Arial CE');

```

### %Przykład 5d - Nasycenie barw w obrazie kolorowym

```

%[L1]=imread('krajobraz1.tif');
clear all; close all; clc;
[L1]=imread('obiekt2.tif');
figure
imshow(L1);
line([12 240],[65 160],'Color',[1 1 1],'LineWidth',3);
c=improfile(L1,[12 240],[65 160]);
%set(gcf,'Color',[1 0 1]);
figure
plot(c(:, :, 1), 'r');
hold on
plot(c(:, :, 2), 'g');
plot(c(:, :, 3), 'b');
set(gcf,'Color',[1 1 1]);
xlabel('k','FontSize',15,'FontName','Arial CE');
ylabel('L(m,n,1)-R, L(m,n,2)-G, L(m,n,3)-B','FontSize',15,'FontName','Arial CE');

```

**Binaryzacja** jest jednym z podstawowych przekształceń stosowanych w analizie obrazów. Proces binaryzacji polega na przekształceniu obrazu mającego wiele poziomów szarości, czyli obrazów monochromatycznych i kolorowych, w obrazy czarno-białe (binarne: biały-1, czarny-0).

**Przykład 6** - Binaryzacja z dwoma dolnymi progami obrazu monochromatycznego (wszystko poniżej progu staje się czarne-0, a powyżej białe-1):

```

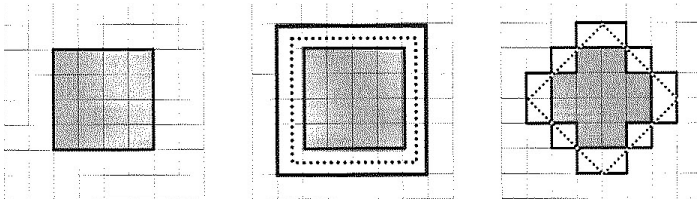
L1=imread('komorki.tif');
figure; imshow(L1);
figure; imhist(L1); grid;
set(gcf,'Color',[1 1 1]);
ylabel('Ilość pikseli','FontSize',15,'FontName','Arial CE')
L2a=L1>210;
figure; imshow(L2a);
L2b=L1>230;
figure; imshow(L2b);

```

Binaryzacja z górnym progami jest negatywem tego samego obrazu binaryzowanego dolnym progami.

### Rozdzielczość przestrzenna obrazu

Na rys. 3 przedstawiono przykładowe problemy występujące podczas powiększenia i obrotu figury na kwadratowej siatce obrazu. Obraz można powiększać tylko skokowo o jeden stopień siatki, na której tworzy się obraz. Pogrubioną linią zaznaczono kontury obiektu powstałego dla dwóch kolejnych możliwych położonych poszczególnych pikseli, natomiast linią przerywaną - położenie figury, gdy występuje nieskończona rozdzielczość optyczna obrazu. W przypadku obrotu obrazu o zadany kąt (w naszym przypadku 45°) pojawia się problem z kształtem otrzymanej figury (obrzeże figury może być utworzone tylko z jednostkowych odcinków położonych tylko w pionie lub w poziomie).



Rys. 3. Graficzna interpretacja powiększenia i obrotu obiektu na kwadratowej siatce obrazu.

### Metody interpolacji przestrzennej obrazu

W przetwarzaniu stopni szarości i intensywności barw punktów obrazu wynikowego na podstawie wartości stopni szarości lub nasycenia barw punktów obrazu źródłowego wykorzystuje się trzy metody interpolacji wartości stopni szarości lub nasycenia barw. Są to:

- metoda najbliższego sąsiada,
- metoda interpolacji dwuliniowej,
- metoda interpolacji dwukwadratowej.

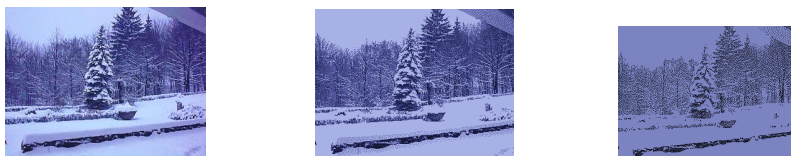
### Przykład 7 - Skutki zmiany rozdzielczości przestrzennej obrazu metodą najbliższego sąsiada:

```
[L1,map]=imread('krajobraz1.tif');
L1=ind2gray(L1,map);
figure; %koniecznie używać figure, nie subplot
imshow(L1,'InitialMagnification','fit');
[L2a]=imresize(L1,size(L1)/2,'nearest');
figure;imshow(L2a,'InitialMagnification','fit');
[L2b]=imresize(L1,size(L1)/4,'nearest');
figure;imshow(L2b,'InitialMagnification','fit');
[L2c]=imresize(L1,size(L1)/8,'nearest');
figure;imshow(L2c,'InitialMagnification','fit');
[L2d]=imresize(L1,size(L1)/16,'nearest');
figure;imshow(L2d,'InitialMagnification','fit');
[L2e]=imresize(L1,round(size(L1)/32),'nearest');
figure;imshow(L2e,'InitialMagnification','fit');
```



### Przykład 8 - Skutki zmiany rozdzielczości poziomów szarości (256->4) z wykorzystaniem funkcji imapprox:

```
[L1,map]=imread('krajobraz1.tif');
figure; imshow(L1,map);
input('');
[L2a,map2a]=imapprox(L1,map,2^8);
figure; imshow(L2a,map2a);
[L2b,map2b]=imapprox(L1,map,2^4);
figure; imshow(L2b,map2b);
[L2c,map2c]=imapprox(L1,map,2^2);
figure; imshow(L2c,map2c);
[L2d,map2d]=imapprox(L1,map,2^1);
figure; imshow(L2d,map2d);
```



**Przetwarzanie dwóch obrazów (dodawanie, odejmowanie, mnożenie, dzielenie).**

Obydwie powyższe operacje, zarówno arytmetyczne jak i logiczne, wykonywane są na odpowiadających sobie punktach dwóch obrazów. Dodatkowo, operacje algebraiczne można wykonywać tylko na dwóch obrazach o tych samych rozmiarach.

W Matlabie dodawanie i odejmowanie dwóch obrazów realizuje się przez dodawanie lub odejmowanie stopnia szarości lub nasycenia barw odpowiadających sobie punktów na dwóch obrazach binarnych, monochromatycznych czy kolorowych.

#### Przykład 9 - Dodawanie dwóch obrazów monochromatycznych:

```
[L1a,map1a] = imread('wieza.tif');
L1a=double(ind2gray(L1a,map1a))/255;
L1a=L1a(273:512,1:320);
figure; imshow(L1a);
[L1b,map1b] = imread('krajobraz1.tif');
L1b=double(ind2gray(L1b,map1b))/255;
% L1b=ind2gray(L1b,map1b);
figure; imshow(L1b)
L2a=L1a+L1b;
figure; imshow(mat2gray(L2a))%metoda suma modulo
figure; imshow(uint8(round(L2a*255)))%metoda suma z nasyceniem
```

#### Przykład 10 - Odejmowanie dwóch obrazów monochromatycznych (obraz "wyjawiony"):

```
clear all; close all; clc;
[L1a,map1a] = imread('wieza.tif');
L1a=ind2gray(L1a,map1a);
L1a=L1a(273:512,1:320)>128;
[L1b,map1b] = imread('krajobraz1.tif');
L1b=ind2gray(L1b,map1b);
L1c=0.05*double(L1a)/255+0.95*double(L1b)/255;
figure(1); imshow(L1c)
title('W tym obrazie schowany jest inny obraz - wyjawimy go')
L2=mat2gray(5*(L1c-0.95*double(L1b)/255));
input('Teraz ukryty obraz zostanie wyjawiony')
figure; imshow(L2)
title('obraz wyjawiony');
```

Mnożenie i dzielenie obrazów stosowane są do polepszenia jakości obrazu.

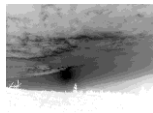
#### Przykład 11 -Poprawienie kontrastu obrazu:

```
clear all; close all; clc;
[L1a,map1a] = imread('wieza.tif');
L1=ind2gray(L1a,map1a);
figure; imshow(L1,[])
L2a=double(L1).^2;
figure; imshow(L2a,[])
```

#### Operacje logiczne.

#### Przykład 12 - Dopełnienie obrazu przy wykorzystaniu funkcji imcomplement:

```
clear all; close all; clc;
[L1a] = imread('krajobraz2.tif');
L1=rgb2gray(L1a);
figure; imshow(L1);
L2=imcomplement(L1);
figure; imshow(L2);
```



### Przykład 13 - Operacja NOT:

```
[L1a,map1a] = imread('wieza.tif');  
L1=ind2gray(L1a,map1a);  
figure('Color','w');imshow(L1);  
colormap(gray(256));  
colorbar  
figure('Color','w');imshow(L1);  
colormap(flipud(gray(256)));  
colorbar
```

### Przykład 14 - Operacja AND:

```
[L1a,map1a] = imread('wieza.tif');  
L1a=ind2gray(L1a,map1a);  
L1a=L1a(273:512,1:320);  
figure;imshow(L1a);  
[L1b,map1b] = imread('krajobraz1.tif');  
L1b=ind2gray(L1b,map1b);  
figure;imshow(L1b);  
L2a=bitand(L1a,L1b);  
figure;imshow(L2a);
```

### Przykład 15 - Operacja OR:

```
[L1a,map1a] = imread('wieza.tif');  
L1a=ind2gray(L1a,map1a);  
L1a=L1a(273:512,1:320);  
figure;imshow(L1a);  
[L1b,map1b] = imread('krajobraz1.tif');  
L1b=ind2gray(L1b,map1b);  
figure;imshow(L1b);  
L2a=bitor(L1a,L1b);  
figure;imshow(L2a);
```

### Wybrane operacje morfologiczne.

Erozja i dylatacja należą do podstawowych przekształceń morfologicznych. Proces erozji na obrazach binarnych można sobie wyobrazić, w najprostszej postaci, jako odcięcie pasa o zadanej szerokości wzdłuż brzegu obiektu. Dylatacja jest procesem odwrotnym.

### Przykład 16 - Zastosowanie gradientu morfologicznego:

```
[L1,map] = imread('krajobraz1.tif');  
L1=ind2gray(L1,map);  
figure; imshow(L1);  
SE=ones(3);  
gradL1=1/2*(imdilate(L1,SE)-imerode(L1,SE));  
figure; imshow(gradL1,[]);
```

Kontur obiektu na obrazie binarnym można wyznaczyć odejmując od obrazu wejściowego obraz, na którym przeprowadzono operację erozji.

Operację związaną z wyznaczeniem gradientu morfologicznego można też rozumieć jako wynik różnicy obrazu powstałego w wyniku przeprowadzenia operacji dylatacji i obrazu pierwotnego.

### Przykład 17 - Wyznaczanie konturu obiektu na obrazie (przy pomocy elementu strukturalnego 3x3):

```
[L1,map] = imread('krajobraz1.tif');
L=ind2gray(L1,map)>128;;
figure; imshow(L);
SE=ones([3,3]);
E=imerode(L,SE);
L2=L-E;
figure; imshow(L2);
```

**Przykład 18** - Morfologiczne wygładzanie.

Operacje otwarcia i zamknięcia są złożeniem erozji i dylatacji. Otwarcie to kolejno wykonane erozja i dylatacja, zamknięcie to kolejno wykonana dylatacja i erozja.

```
[L1,map] = imread('krajobraz1.tif');
L1=ind2gray(L1,map);
figure; imshow(L1);
SE=ones(3);
L2=imclose(imopen(L1,SE),SE);
figure; imshow(L2);
```