



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Przetwarzanie obrazów 2020/2021

Wykład nr 2

Dr inż. Przemysław Korohoda
E-mail: korohoda@agh.edu.pl
Tel.wewn.AGH: (012-617)-27-52
Pawilon C3 - p.506

Strona WWW:
home.agh.edu.pl/~korohoda/rok_2020_2021_zima/PO_IA_4_WIMiR

Plan wykładu

- 1. Studium modelu HSV**
- 2. Model kamery**
- 3. Rejestracja obrazów**
- 4. Zmiana rozdzielczości**
- 5. Histogram**

Studium modelu HSV

```
clc; clear; close all;
```

```
N=60;    N1=N/2+0.5;    N2=N/2;
v=1.0; % value - czyli mniej wiecej: "jasnosc";
Pmask(1:N,1:N)=0;
```

```
for m=1:N
```

```
    for n=1:N
```

```
        if sqrt( (m-N1)^2+(n-N1)^2 ) < N2,
            Phsv(m,n,1) = atan2( (m-N1), (n-N1) ) / (2*pi) + 0.5;
            Phsv(m,n,2) = sqrt( (m-N1)^2+(n-N1)^2 ) / N2;
            Phsv(m,n,3) = v;
            Pmask(m,n) = 1;
```

```
        end;
```

```
    end;
```

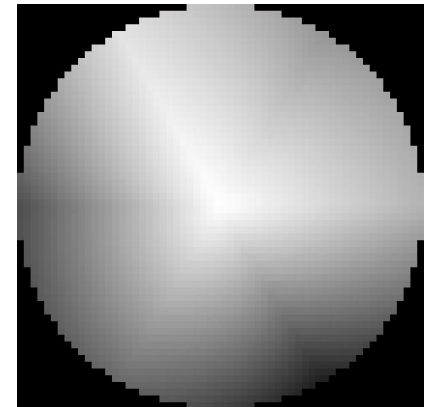
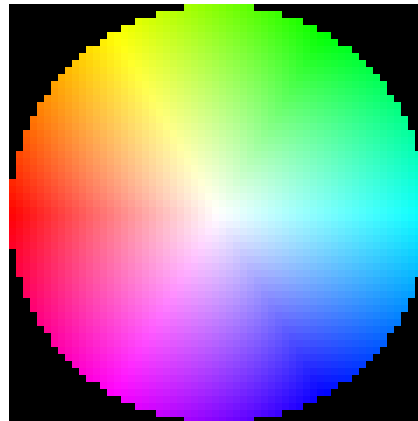
```
end;
```

```
Phsv(:, :, 1)=Pmask.*Phsv(:, :, 1);
```

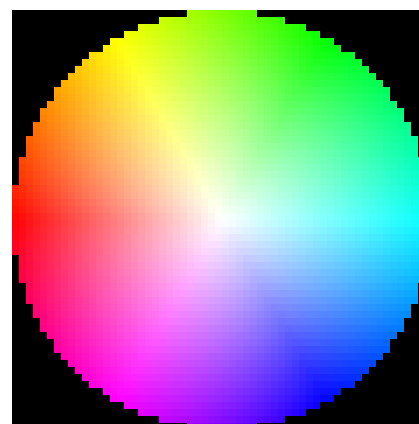
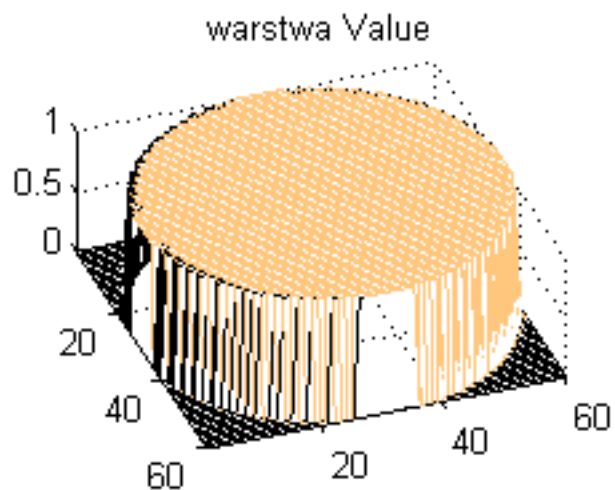
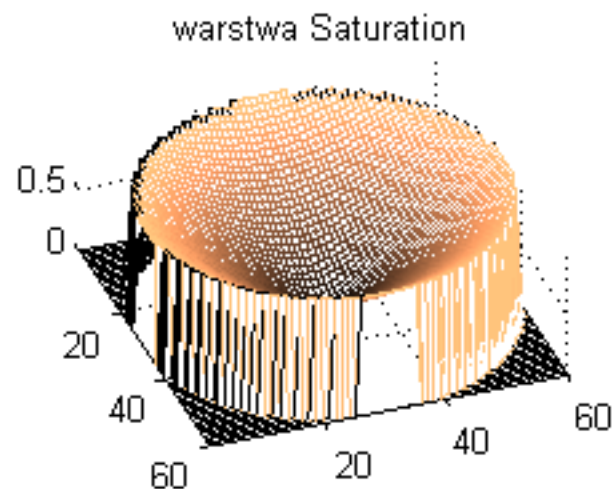
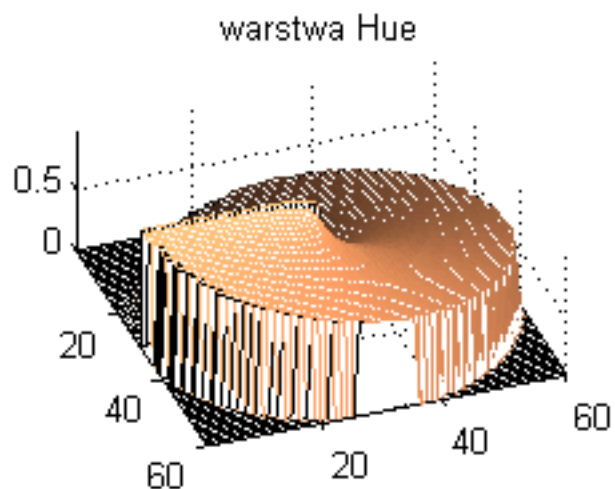
```
Phsv(:, :, 2)=Pmask.*Phsv(:, :, 2);
```

```
Phsv(:, :, 3)=Pmask.*Phsv(:, :, 3);
```

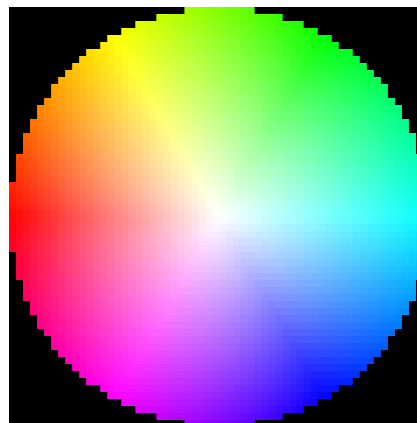
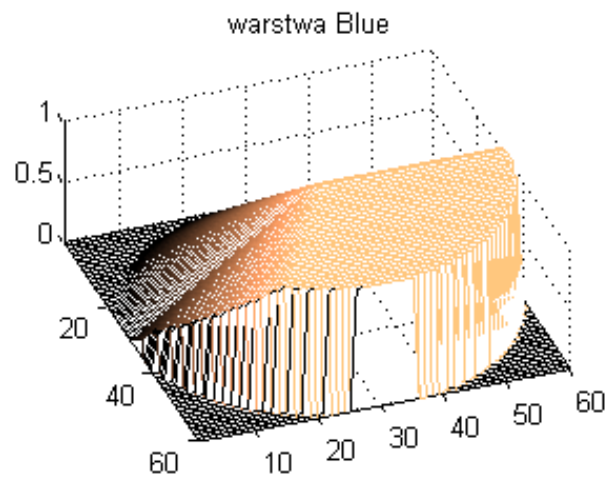
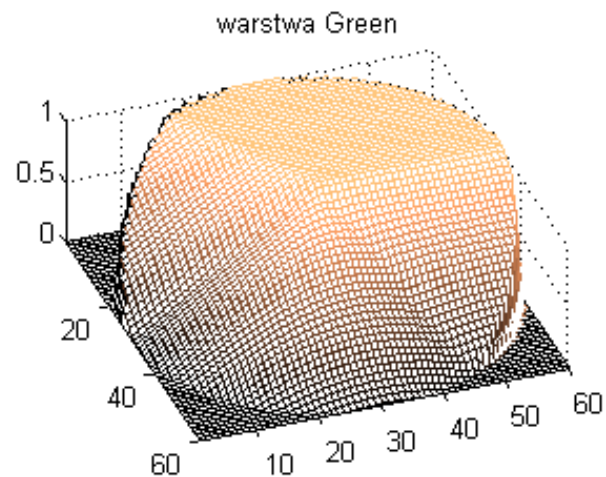
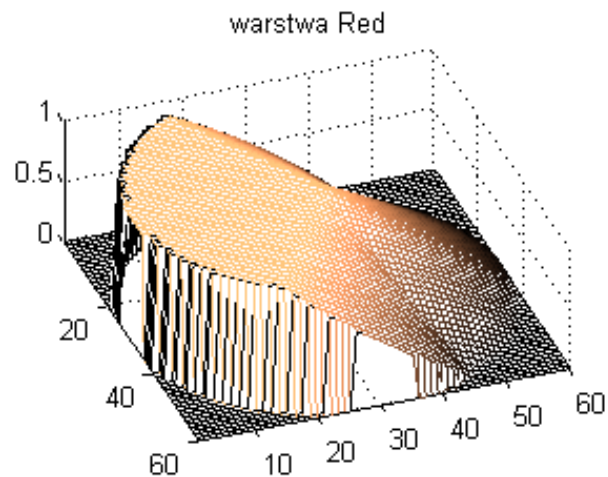
```
Prgb=hsv2rgb(Phsv);
```



Studium modelu HSV



Studium modelu HSV



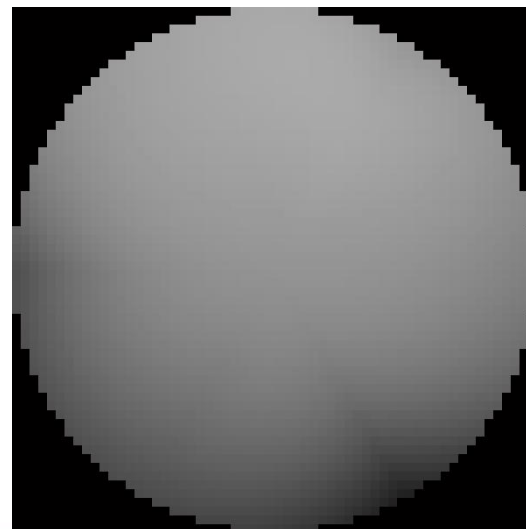
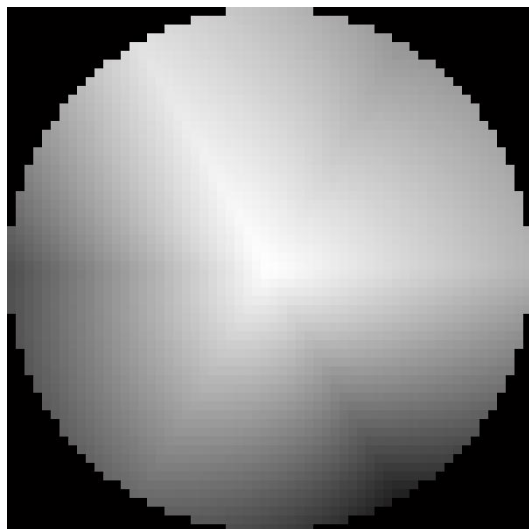
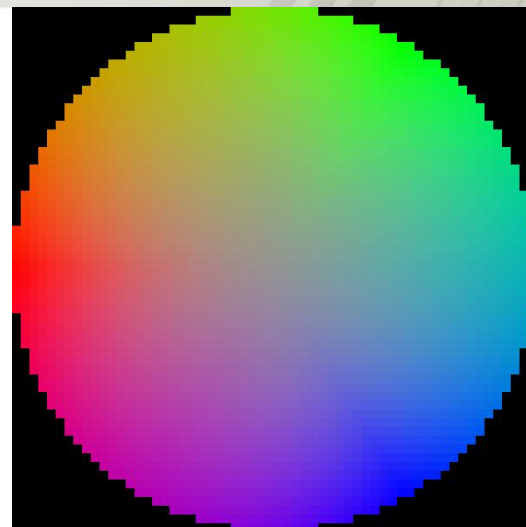
Wprowadzenie normalizacji (korekty)

```
korekta=1;
switch korekta,
    case 0,    dzielnik_Prgb = 1;
    case 1,    norma_Prgb = sqrt( Prgb(:, :, 1).^2 + Prgb(:, :, 2).^2 + Prgb(:, :, 3).^2 );
               dzielnik_Prgb = norma_Prgb / v;
end;

Prgb(:, :, 1)=Pmask.*Prgb(:, :, 1)./dzielnik_Prgb;
Prgb(:, :, 2)=Pmask.*Prgb(:, :, 2)./dzielnik_Prgb;
Prgb(:, :, 3)=Pmask.*Prgb(:, :, 3)./dzielnik_Prgb;

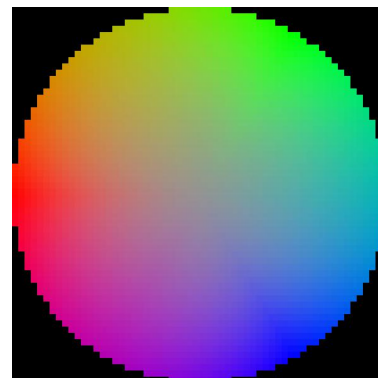
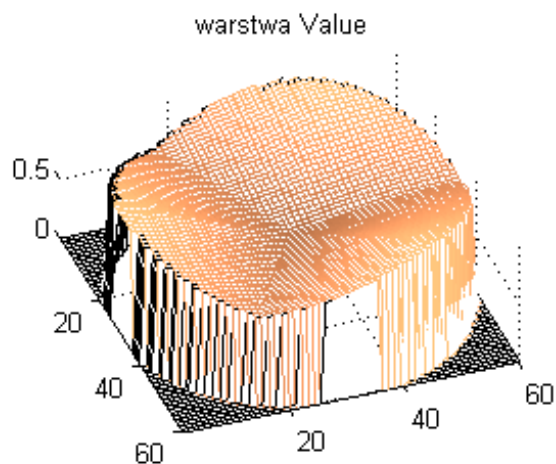
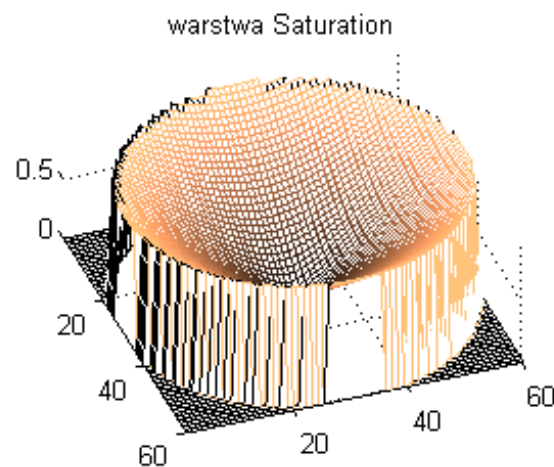
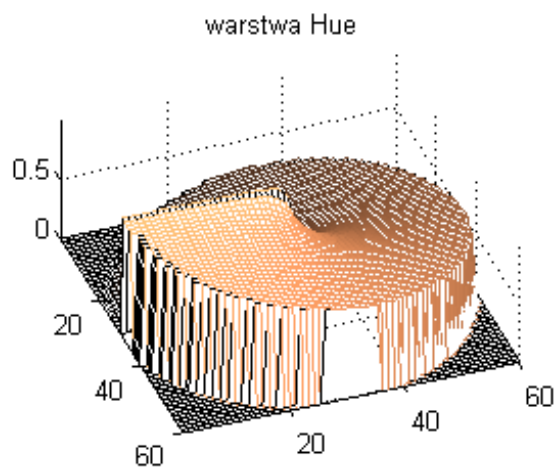
Phsv = rgb2hsv(Prgb);
Pgray = rgb2gray(Prgb);
```

Studium modelu HSV - normalizacja



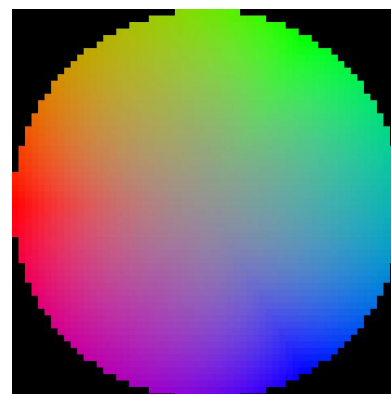
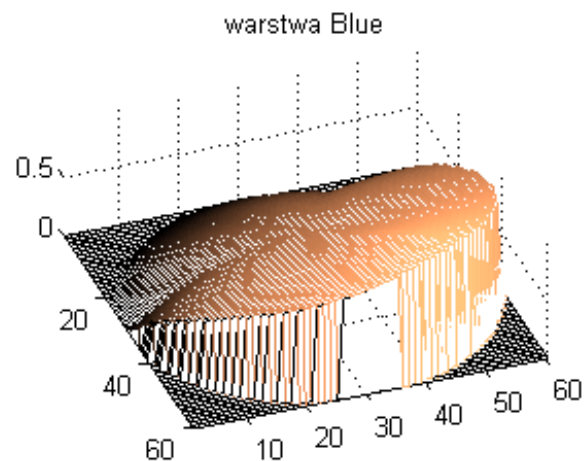
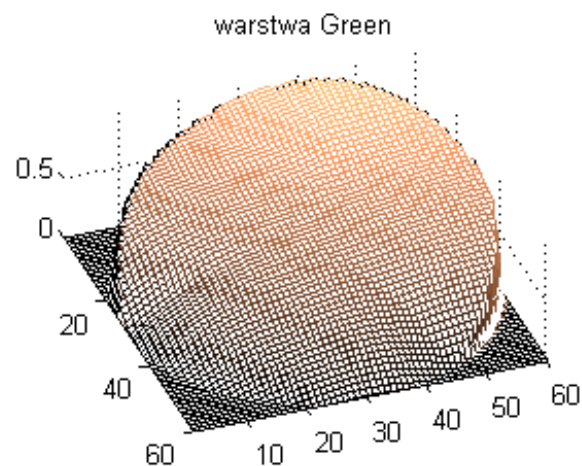
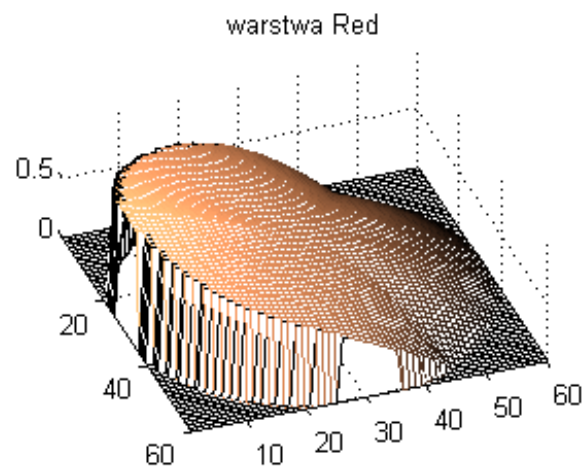
Studium modelu HSV

Wynik z korektą (normalizacją)



Studium modelu HSV

Wynik z korektą (normalizacją)



Czułość oka dla R, G, B

Prgb(1:3,1:3,1)=1; Prgb(1:3,1:3,2)=0; Prgb(1:3,1:3,3)=0;

Pgray=rgb2gray(Prgb),

Pgray =

0.2989	0.2989	0.2989
0.2989	0.2989	0.2989
0.2989	0.2989	0.2989

Prgb(1:3,1:3,1)=0; **Pr**gb(1:3,1:3,2)=1; Prgb(1:3,1:3,3)=0;

Pgray=rgb2gray(Prgb),

Pgray =

0.5870	0.5870	0.5870
0.5870	0.5870	0.5870
0.5870	0.5870	0.5870

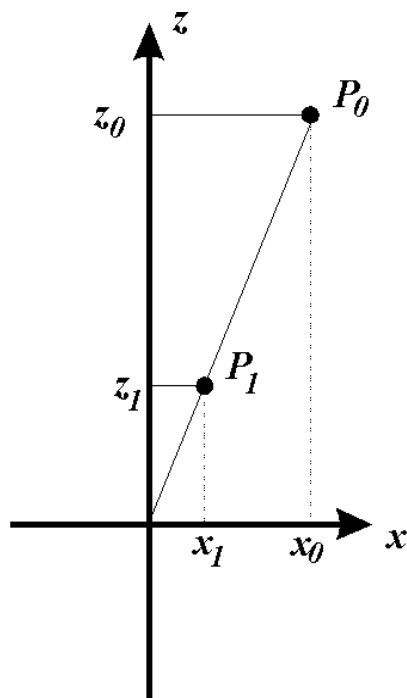
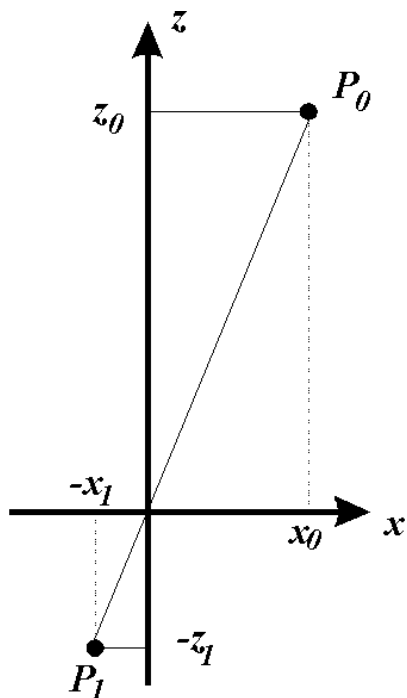
Prgb(1:3,1:3,1)=0; Prgb(1:3,1:3,2)=0; **Pr**gb(1:3,1:3,3)=1;

Pgray=rgb2gray(Prgb),

Pgray =

0.1140	0.1140	0.1140
0.1140	0.1140	0.1140
0.1140	0.1140	0.1140

Model kamery z rzutem perspektywicznym (*pin-hole*)



$$\frac{F}{x_{pix}} = \frac{z_1}{x_1} = \frac{z_0}{x_0}$$

Ogniskowa:

$$F = \frac{z_0}{x_0} \cdot x_{pix}$$

Dwa warianty

Organizacja matryc CCD

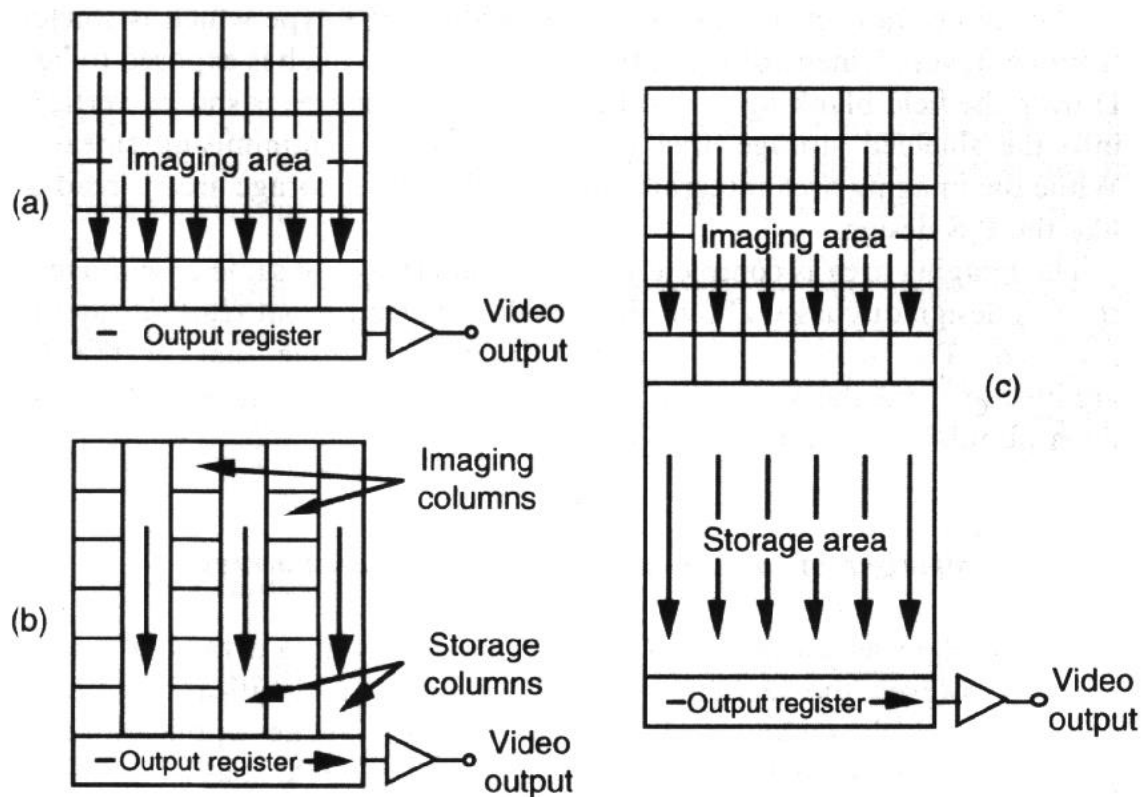


Figure 3.12 Schematic representations of CCD area array architectures: (a) parallel/serial; (b) interline transfer; (c) frame transfer. After Batchelor *et al.* [5]

Zmiana rozdzielczości obrazu

Interpolacja liniowa

Interpolacja liniowa

$$\begin{cases} y(x_0) = a \cdot x_0 + b \\ y(x_1) = a \cdot x_1 + b \end{cases} \Rightarrow y(x) = a \cdot x + b$$

Interpolacja dwuliniowa (*bilinear*)

dla $x_0 = 0; \quad x_1 = 1; \quad y_0 = 0; \quad y_1 = 1 :$

$$\begin{cases} P(x,0) = (P(1,0) - P(0,0)) \cdot x + P(0,0) \\ P(x,1) = (P(1,1) - P(0,1)) \cdot x + P(0,1) \end{cases}$$

$\Downarrow$

$$P(x,y) = (P(x,1) - P(x,0)) \cdot y + P(x,0)$$

Interpolacja (bi)kubiczna/(dwu)sześcienna (*bicubic*)

Dla ustalonej wartości y tworzymy i rozwiązujemy równanie macierzowe:

$$\begin{cases} P(x_0, y) = a \cdot x_0^3 + b \cdot x_0^2 + c \cdot x_0 + d \\ P(x_1, y) = a \cdot x_1^3 + b \cdot x_1^2 + c \cdot x_1 + d \\ P(x_2, y) = a \cdot x_2^3 + b \cdot x_2^2 + c \cdot x_2 + d \\ P(x_3, y) = a \cdot x_3^3 + b \cdot x_3^2 + c \cdot x_3 + d \end{cases} \Rightarrow P(x, y) = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$$

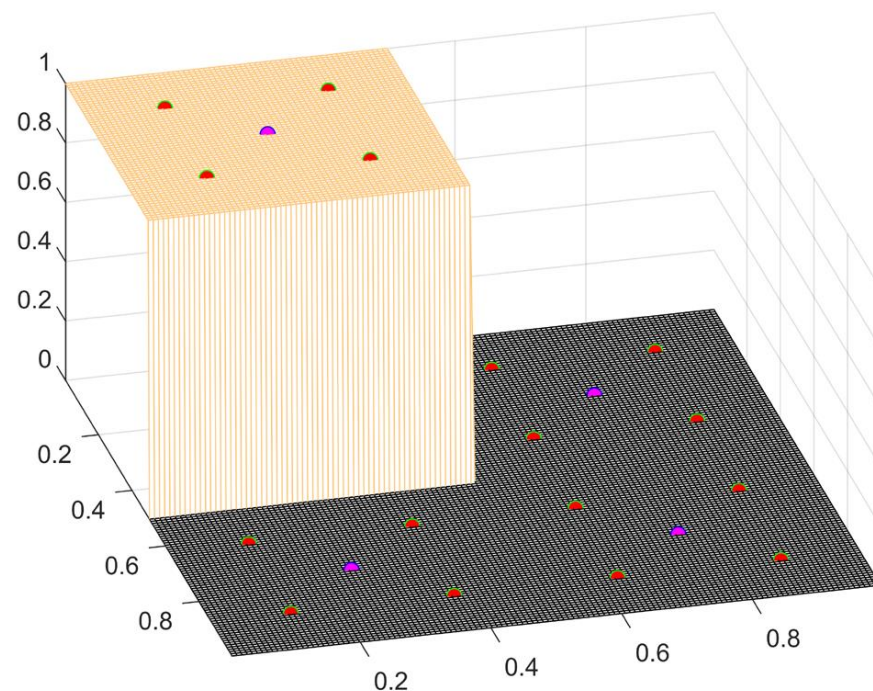
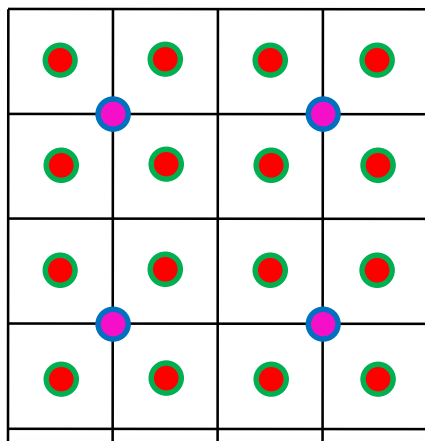
**W bicubic – analogicznie, ale w dziedzinie dwuwymiarowej,
zatem konieczne są dwa kroki:
drugi dla czterech wartości y oraz ustalonej wartości x .**

Przykłady interpolacji obrazu

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

P1=imresize(P,2,'nearest');

$$P_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

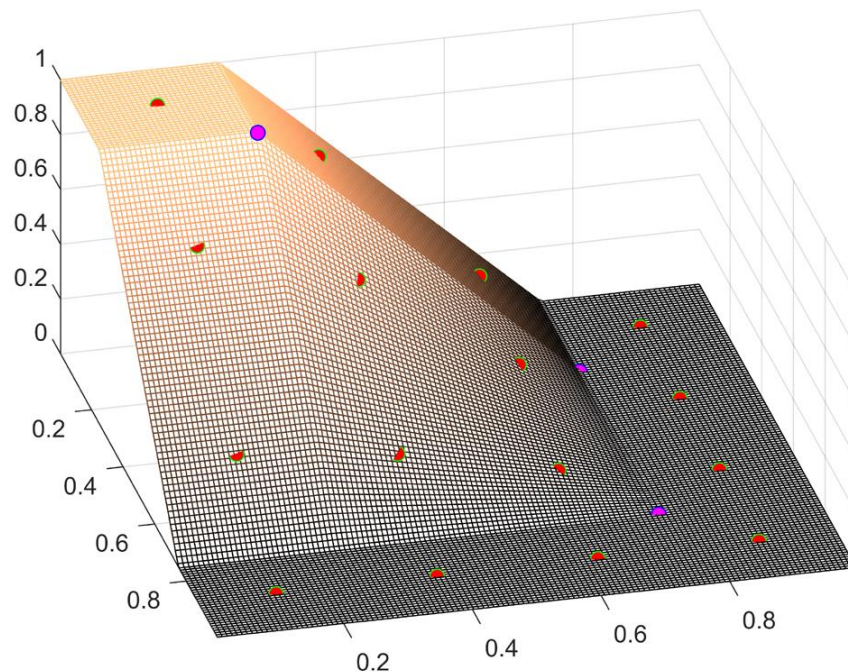
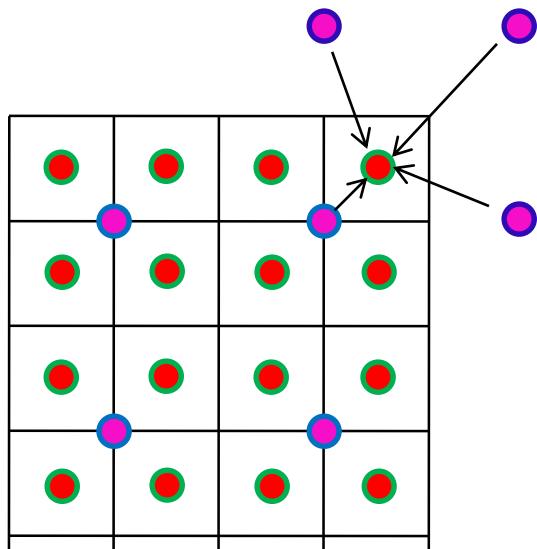


Przykłady interpolacji obrazu

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

P2=imresize(P,2,'bilinear');

$$P_2 = \begin{bmatrix} 1 & 12/16 & 4/16 & 0 \\ 12/16 & 9/16 & 3/16 & 0 \\ 4/16 & 3/16 & 1/16 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

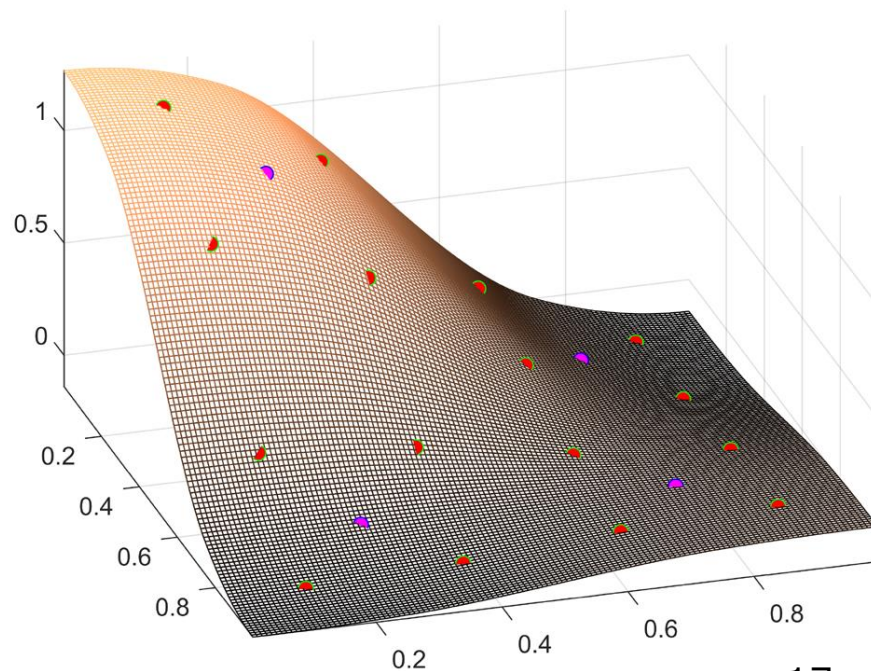
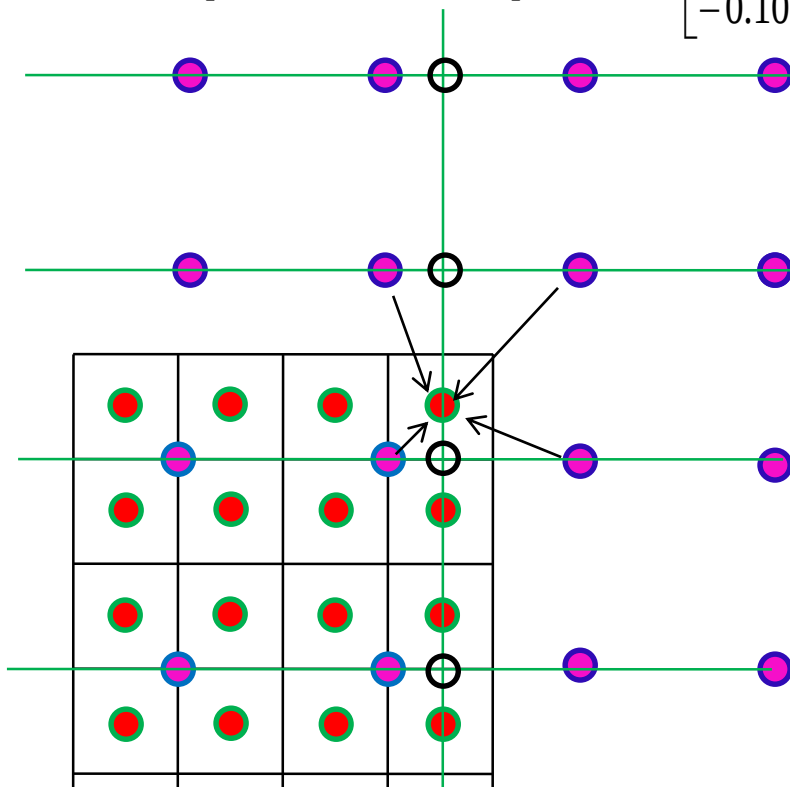


Przykłady interpolacji obrazu

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

P3=imresize(P,2,'bicubic');

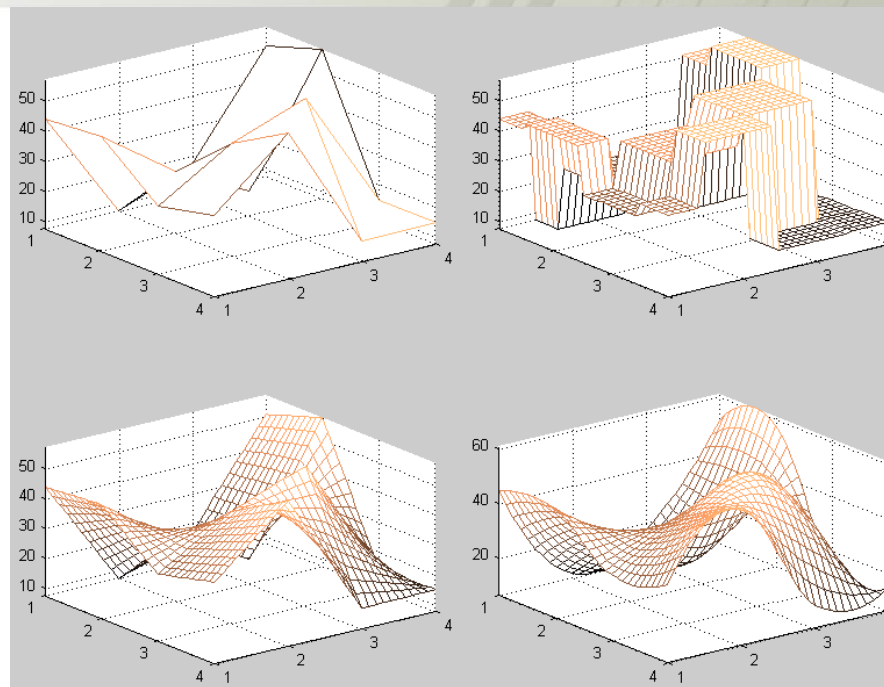
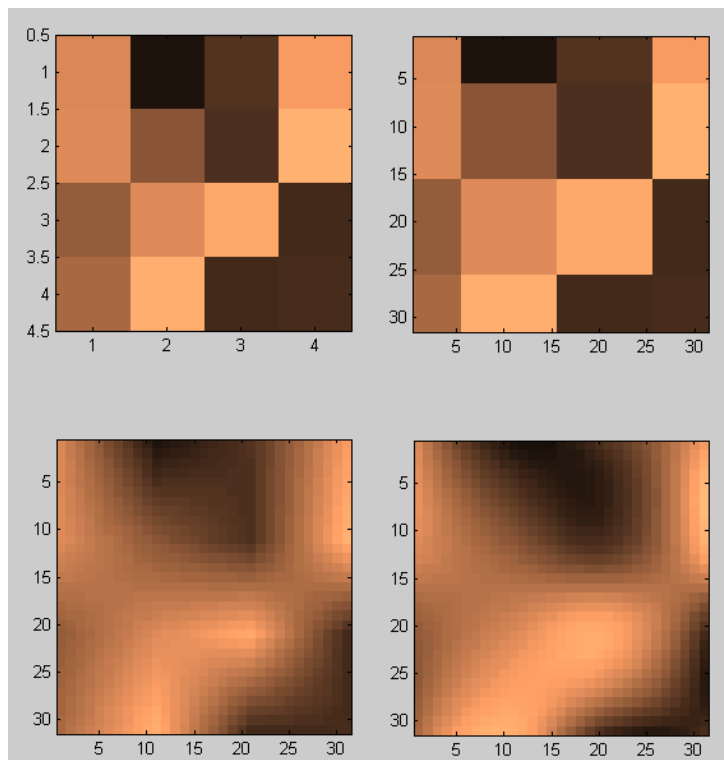
$$P_3 = \begin{bmatrix} 1.1963 & 0.8716 & 0.2222 & -0.1025 \\ 0.8716 & 0.6350 & 0.1619 & -0.0747 \\ 0.2222 & 0.1619 & 0.0413 & -0.0190 \\ -0.1025 & -0.0747 & -0.0190 & 0.0088 \end{bmatrix}$$



Przykłady interpolacji obrazu

**Obraz 4x4
powiększony 8 razy
- trzy metody interpolacji**

Obrazy

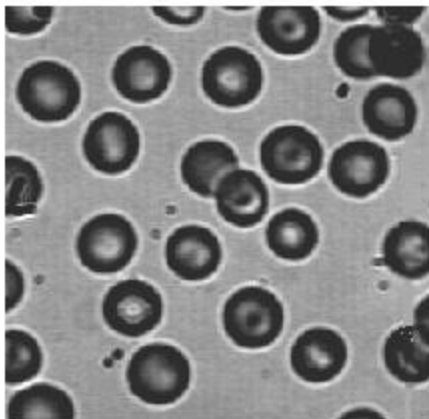


Wykresy obrazów

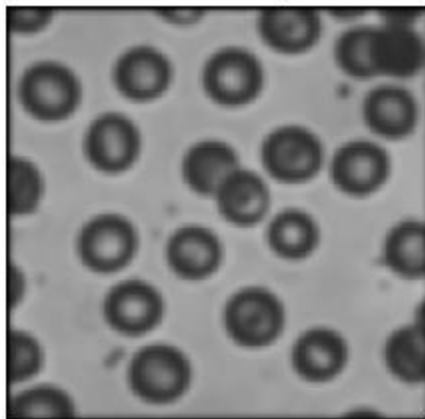
Interpolacja (Matlab: imresize)

Po 100-krotnej zmianie skali (x3, x1/3), interpolacja bikubiczna:

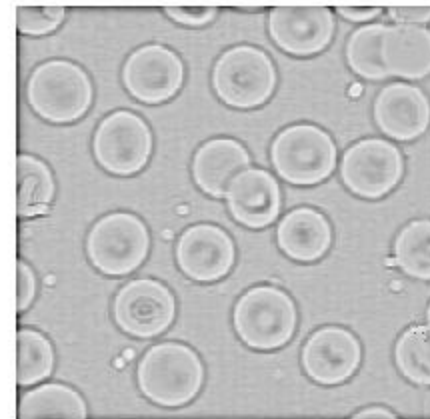
Oryginal



Odtworzony



Obraz błędu



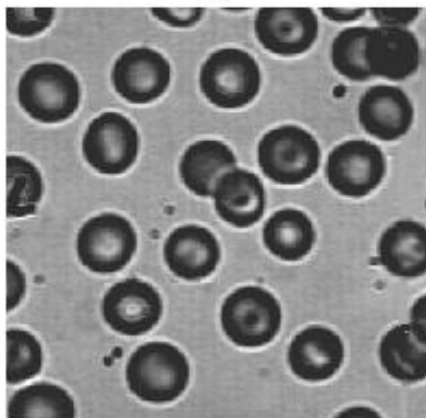
$$P_{error} = P_{odtw} - P_{oryg}$$

Uwaga na zakres wartości pikseli

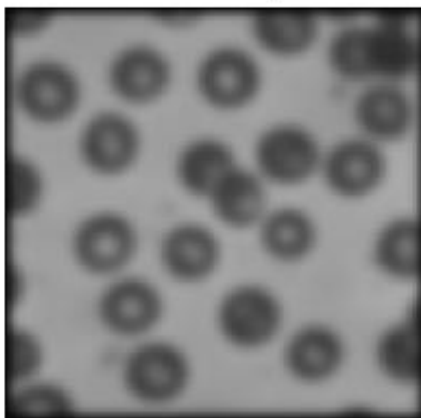
Interpolacja (Matlab: imresize)

Po 100-krotnej zmianie skali ($\times 3$, $\times 1/3$), interpolacja biliniowa:

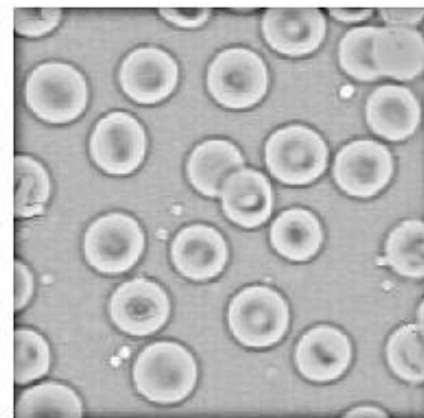
Oryginal



Odtworzony



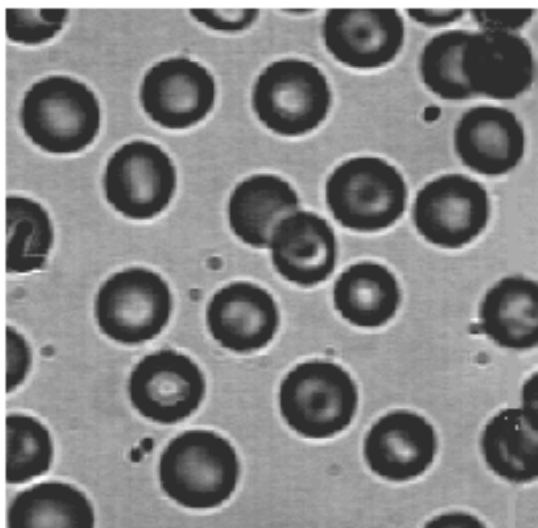
Obraz błedu



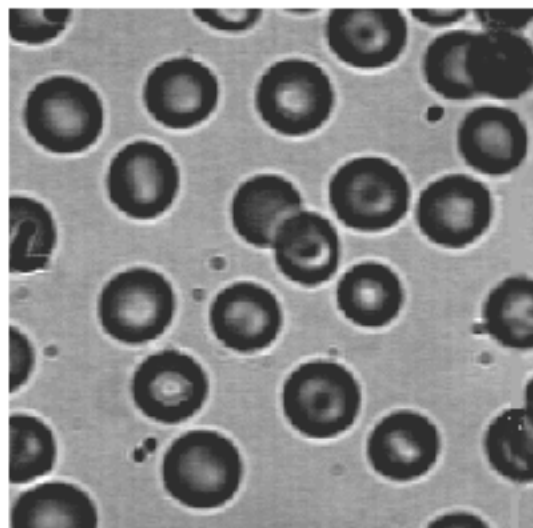
Interpolacja (Matlab: imresize)

Po 100-krotnej zmianie skali ($\times 3$, $\times 1/3$), interpolacja według „najbliższego sąsiada”:

Oryginal



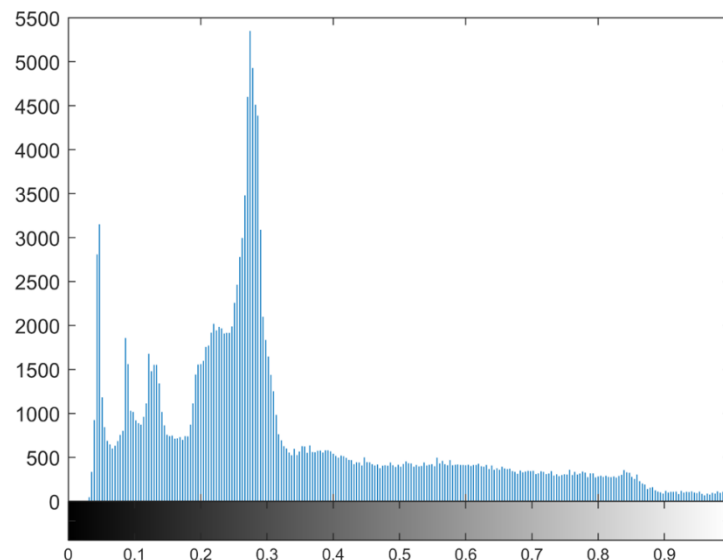
Odtworzony



Histogram – wizualizacja statystycznego rozkładu wartości pikseli

**Obraz monochromatyczny,
8-bitowy**

```
.....  
P=imread(nazwa);  
P=rgb2gray(P);  
P=double(P)/256;  
  
figure(1); clf;  
imhist(P);  
  
.....
```

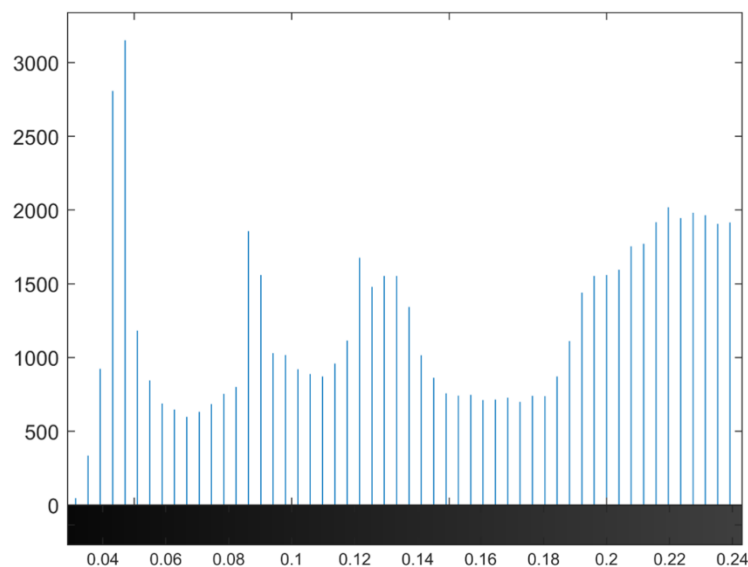
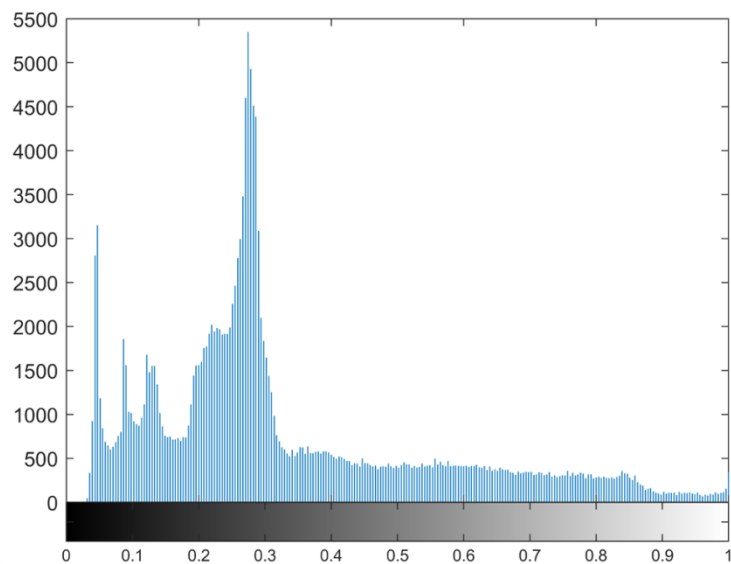


Histogram (cd.)



Po użyciu imhist

Powiększony fragment:



Histogram (cd.)

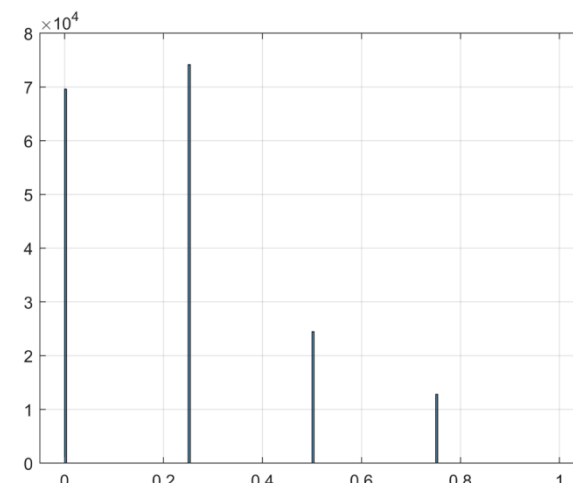
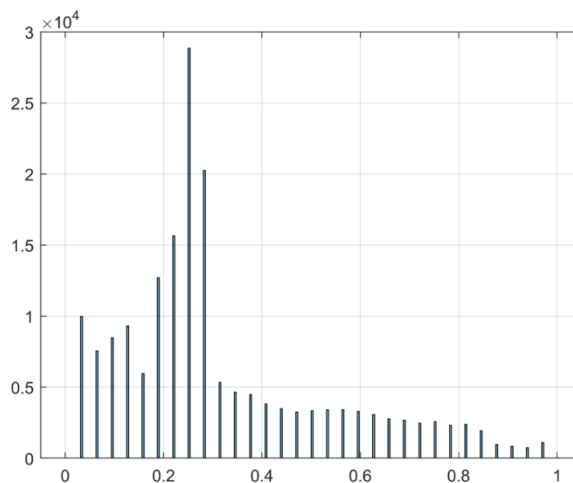
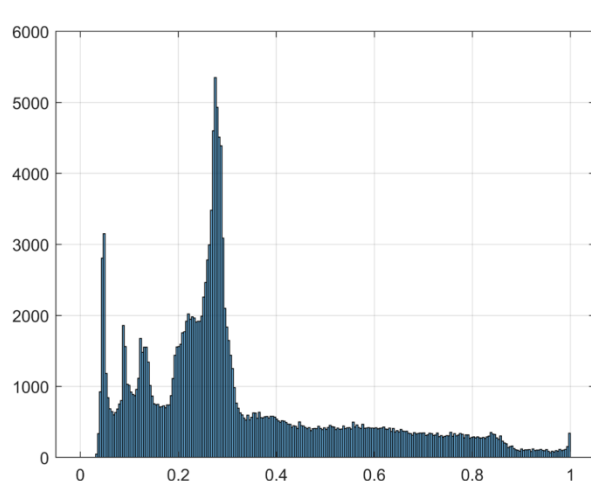
256 poziomów



32 poziomy



4 poziomy



Histogramy po użyciu funkcji **histogram**, dla 256 przedziałów

Histogram (cd.)

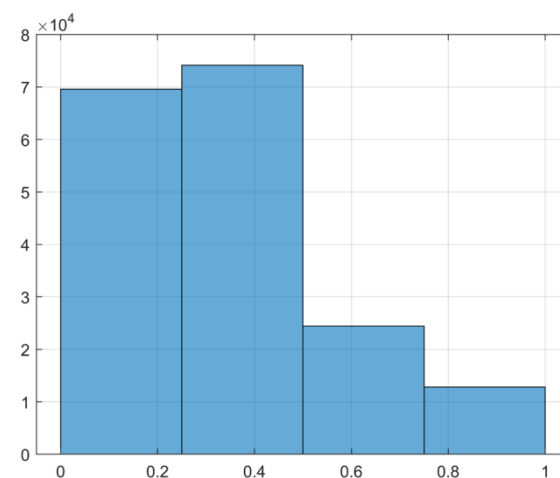
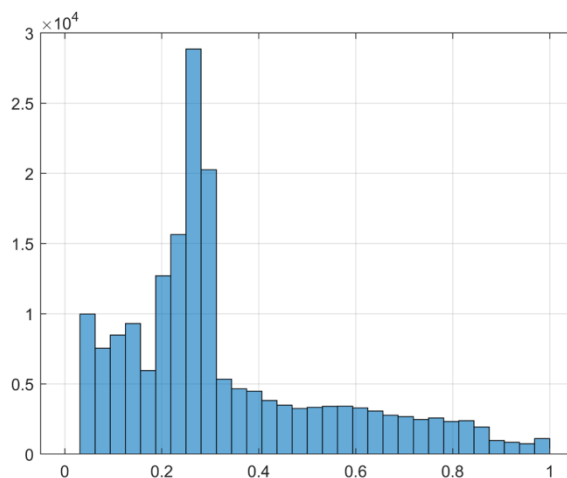
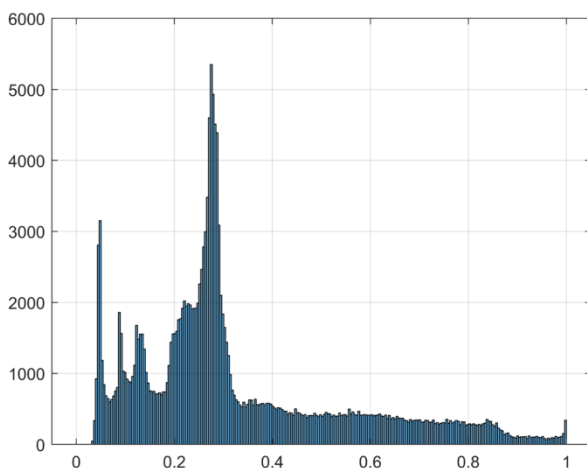
256 poziomów



32 poziomy



4 poziomy



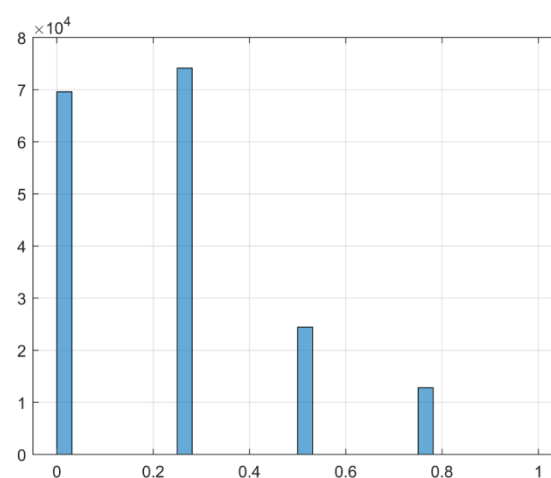
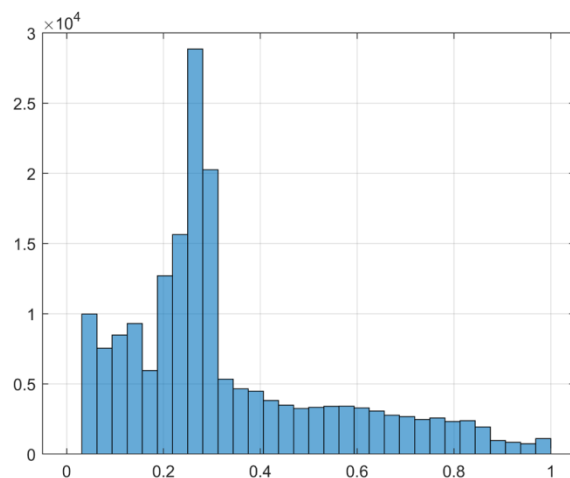
Po użyciu funkcji **histogram**, dla 256, 32 oraz 4 przedziałów

Histogram (cd.)

32 poziomy



4 poziomy



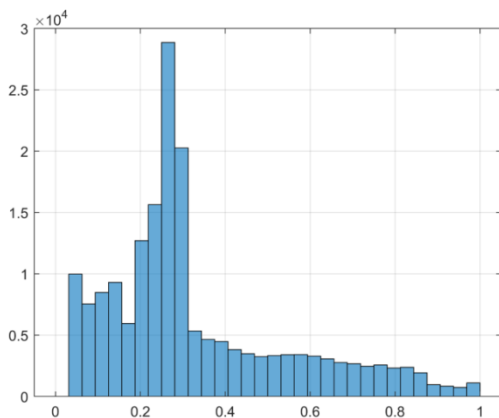
Po użyciu funkcji **histogram**, dla 32 przedziałów

Histogram (cd.)

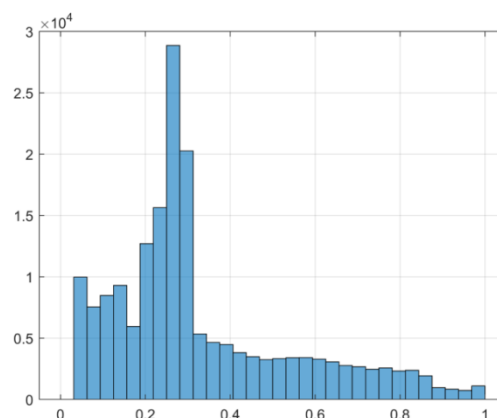
```
.....  
P=imread(nazwa);  
P=rgb2gray(P);  
P=double(P); P=floor(P/64)*64; % teraz: 4 poziomy  
  
    figure(7); clf;  
        histogram(P,0:1/256:1); grid on;  
        %histogram(P,0:1/32:1); grid on;  
        %histogram(P,0:1/4:1); grid on;  
  
.....
```

Histogram (cd.)

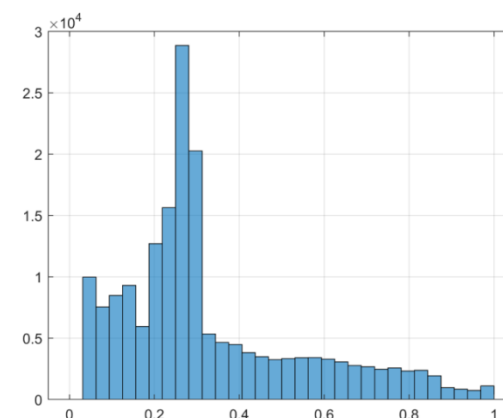
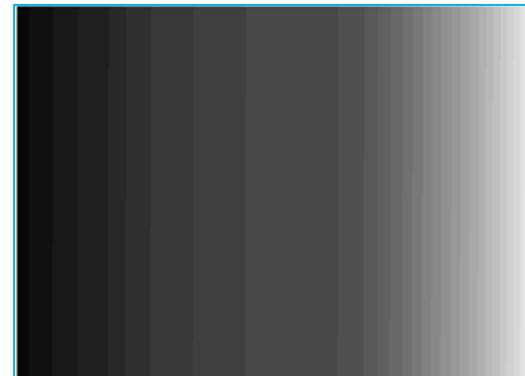
32 poziomy



**przetrasowane
pseudolosowo
piksele**



**posortowane
piksele**



Po użyciu funkcji **histogram**, dla 32 przedziałów

Histogram (cd.)

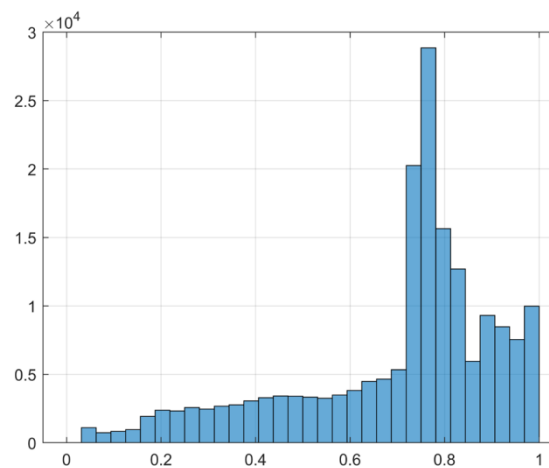
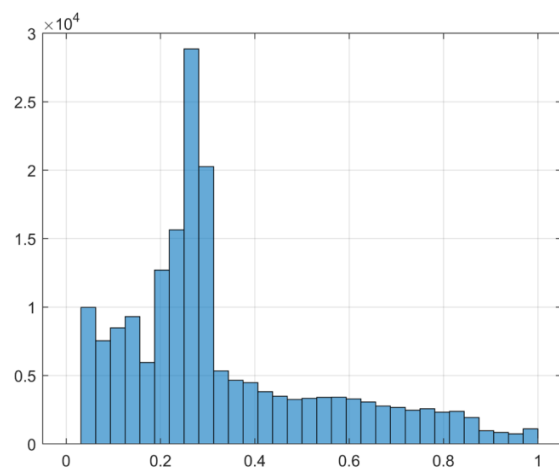
```
.....  
operacja=1;  
switch operacja  
    case 1, % nic nie robimy;  
    case 2, P=1-P; % negacja;  
    case 3, % losowe przetasowanie;  
        [M,N]=size(P); k=randperm(M*N); P=P(:); P=P(k);  
        P=reshape(P,[M,N]);  
    case 4, % ułożenie kolejno (sortowanie);  
        [M,N]=size(P); P=P(:); P=sort(P);  
        P=reshape(P,[M,N]);  
end  
.....
```

Histogram (cd.)

32 poziomy



negatyw



Po użyciu funkcji **histogram**, dla 32 przedziałów

Histogram dla obrazów RGB

W przypadku obrazów barwnych (np. RGB) wydzielamy każdy z kanałów i dla każdego wyznaczamy osobny histogram.

Warto może skorzystać z funkcji `subplot(a,b,c)`.

Wybrane „obrazowe” i inne ważne funkcje Matlaba

imresize

P2=imresize(P1,skala);

P2=imresize(P1,skala, interpolacja); *interpolacja='nearest', 'bilinear', 'bicubic';
default: interpolacja = 'nearest' (?)*

P2=imresize(P1,[wiersze,kolumny]);

dla skala<1:

P2=imresize(P1,skala,N); **N - długość boku pola filtru antyaliasingowego;**
default: N=11;

P2=imresize(P1,skala,H); **H - maska filtru antyaliasingowego;**

Warto poznać także: **imrotate, **imcrop**.**

Przykład pliku

```
function Prgb=Ppasy(wariant,formatp);
% Prgb=Ppasy(wariant,formatp);
%
% Przykłady tworzenia obrazu kolorowych
%          pasow;
%
% wariant=1 - tworzymy najpierw obraz
%          indeksowany;
% wariant=2 - tworzymy bezpośrednio obraz
%          RGB;
% formatp=1 - double;
% formatp=2 - uint8;
%
% (C) P.Korohoda

% a) obraz indeksowany:
if wariant==1,
    P=ones(8,1)*(1:8);
    mapa=[1,0,0;0,1,0;0,0,1;0,1,1;1,0,1;1,1,0;;1,1,1;0,0,0];
    Prgb=ind2rgb(P,mapa);
end;
```

```
% b) obraz RGB:
if wariant==2,
    Prgb(1:8,1:8,1)=ones(8,1)*[1,0,0,0,1,1,1,0];
    Prgb(1:8,1:8,2)=ones(8,1)*[0,1,0,1,0,1,1,0];
    Prgb(1:8,1:8,3)=ones(8,1)*[0,0,1,1,1,0,1,0];
end;

% ewentualna konwersja formatu liczb:
if formatp==2,
    Prgb=uint8(round(Prgb*255));
end;

% KONIEC PLIKU
```

Zapraszam do laboratorium ...