PODSTAWY PRZETWARZANIA OBRAZÓW CYFROWYCH

Przekształcenia kontekstowe

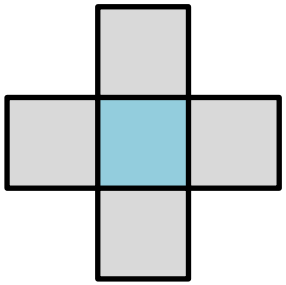
adam.piorkowski@agh.edu.pl

Przekształcenia kontekstowe

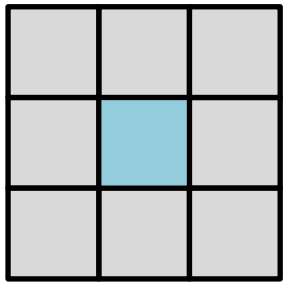
- Sąsiedztwo piksela
- Filtry liniowe
  - filtry splotowe (konwolucyjne)
    - kierunkowe, uśredniające, wyostrzające, dolnoprzepustowe/górnoprzepustowe,
    - filtry Robertsa, Prewitta, Sobela, laplasjany, Gaussa,
- Kombinacje filtrów
  - Laplacian of Gaussian, Difference of Gaussian,
    - Mexican Hat (blob detection)
- Filtry nieliniowe
  - filtry rankingowe
    - mediana, min, max
- Inne operacje kontekstowe
  - AHE (CLAHE) ….

Sąsiedztwo piksela

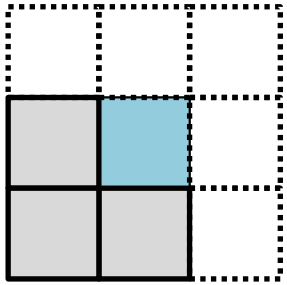
złożoność obliczeniowa!



4-connected

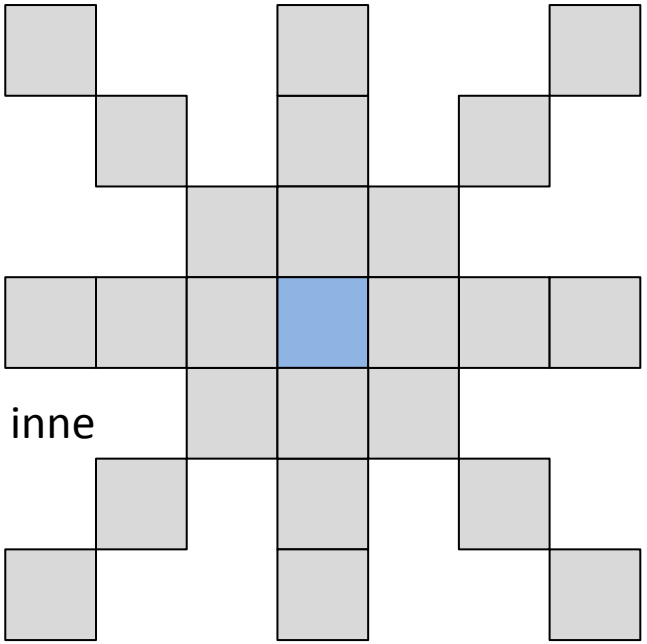


8-connected
3x3

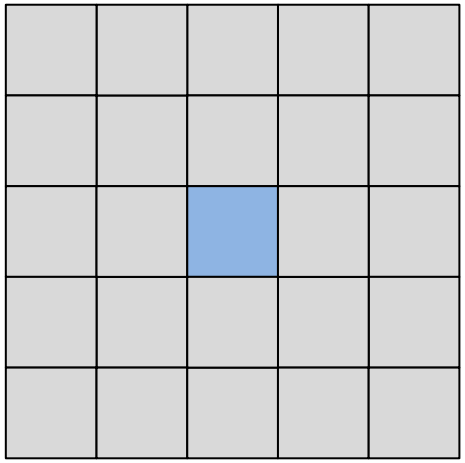


2x2
niezalecane

O (4r)

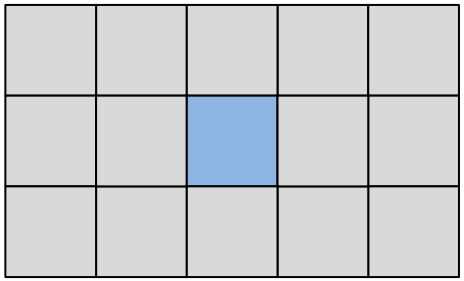


inne



kwadrat
5x5

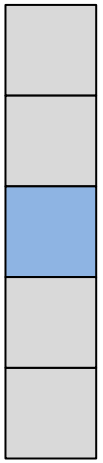
O(r²)



prostokąt
5x3

*O (r_a*r_b)*

f. kierunkowe

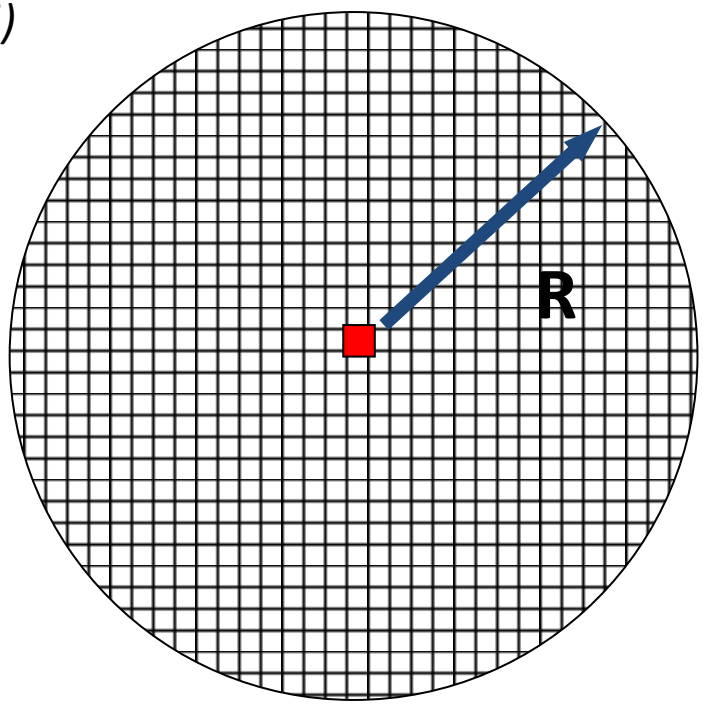


O (r)

Sąsiedztwo piksela

$rx^2 + ry^2 <= R^2$

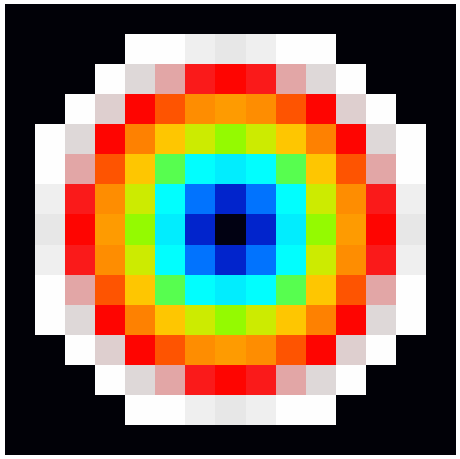
$O(r^2)$



5.7	5.0	4.5	4.1	4.0	4.1	4.5	5.0	5.7
5.0	4.2	3.6	3.2	3.0	3.2	3.6	4.2	5.0
4.5	3.6	2.8	2.2	2.0	2.2	2.8	3.6	4.5
4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1
4.0	3.0	2.0	1.0	**R**	1.0	2.0	3.0	4.0
4.1	3.2	2.2	1.4	1.0	1.4	2.2	3.2	4.1
4.5	3.6	2.8	2.2	2.0	2.2	2.8	3.6	4.5
5.0	4.2	3.6	3.2	3.0	3.2	3.6	4.2	5.0
5.7	5.0	4.5	4.1	4.0	4.1	4.5	5.0	5.7

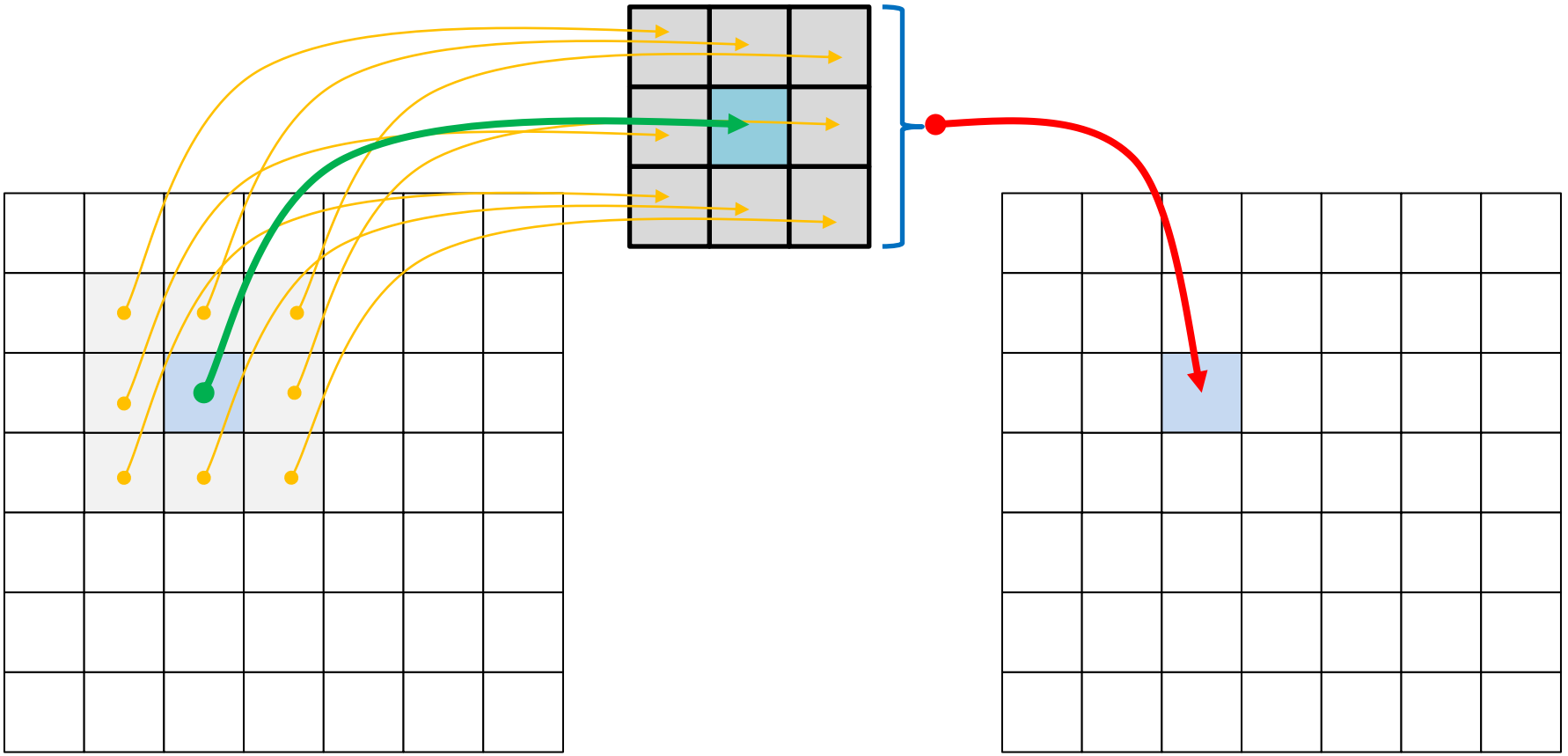
zaokrąglone do 0.1 !

$1.4 < 1.41 < \sqrt{2}$



Rozmiar maski istotny – dla obrazu o innym rozmiarze (powiększonego, pomniejszonego) ten sam filtr może działać inaczej

Przekształcenia kontekstowe



0	0	0	0	0	0	0	0	10	90	90	90	90	90	90	90
0	0	0	0	0	0	0	10	80	90	90	90	90	90	90	90
0	0	10	20	0	0	10	80	90	90	90	90	80	0	90	90
0	0	20	10	0	10	80	90	90	90	80	0	0	20	90	90
0	0	10	0	0	80	80	90	90	80	10	0	0	0	90	90
0	0	0	0	20	80	90	90	90	80	20	0	0	0	90	90
0	10	0	0	80	90	90	90	90	80	80	0	0	80	90	90
0	0	0	80	90	90	90	90	90	90	20	80	0	10	90	90
0	0	20	90	90	90	90	90	90	80	20	80	80	0	90	90
0	0	80	90	90	90	90	90	90	30	90	90	90	90	90	90
0	80	90	90	90	20	90	90	90	90	90	90	90	90	90	90
90	90	90	90	20	20	20	90	90	90	90	90	90	90	90	90
90	90	90	20	20	10	20	30	90	90	90	90	70	80	90	90
90	90	90	80	20	20	20	30	90	90	90	90	60	70	80	90
90	90	90	90	20	30	20	80	90	90	90	90	90	90	80	90
90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

r=3.75, t=50

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

suma = 45

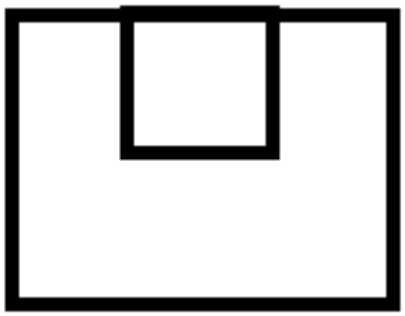
0	0	0	0	2	5	9	14	20	0	0	0	0	0	0	0
0	0	0	2	5	9	14	20	0	0	0	0	0	0	0	0
0	0	1	4	8	13	19	0	0	0	0	0	0	34	0	0
0	0	3	7	12	18	0	0	0	0	0	31	31	31	0	0
0	2	6	11	17	0	0	0	0	0	29	28	28	29	0	0
1	4	9	15	21	0	0	0	0	0	28	26	26	27	0	0
3	7	13	19	0	0	0	0	0	0	0	26	27	0	0	0
6	11	17	0	0	0	0	0	0	0	30	0	29	31	0	0
11	16	20	0	0	0	0	0	0	0	34	0	0	35	0	0
15	19	0	0	0	0	0	0	0	38	0	0	0	0	0	0
21	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0
0	0	0	0	26	29	28	0	0	0	0	0	0	0	0	0
0	0	0	27	29	29	29	29	0	0	0	0	0	0	0	0
0	0	0	0	29	29	29	30	0	0	0	0	0	0	0	0
0	0	0	0	30	30	30	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

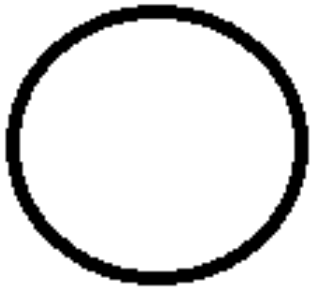
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

b=25

Nurzynska, K., Mikhalkin, A., Piorkowski, A.:
CAS: Cell Annotation Software - Research on Neuronal Tissue Has Never Been so Transparent.
Neuroinformatics, 2017, Vol. 15, Iss. 4, pp 365-382.

Sąsiedztwo - artefakty





+

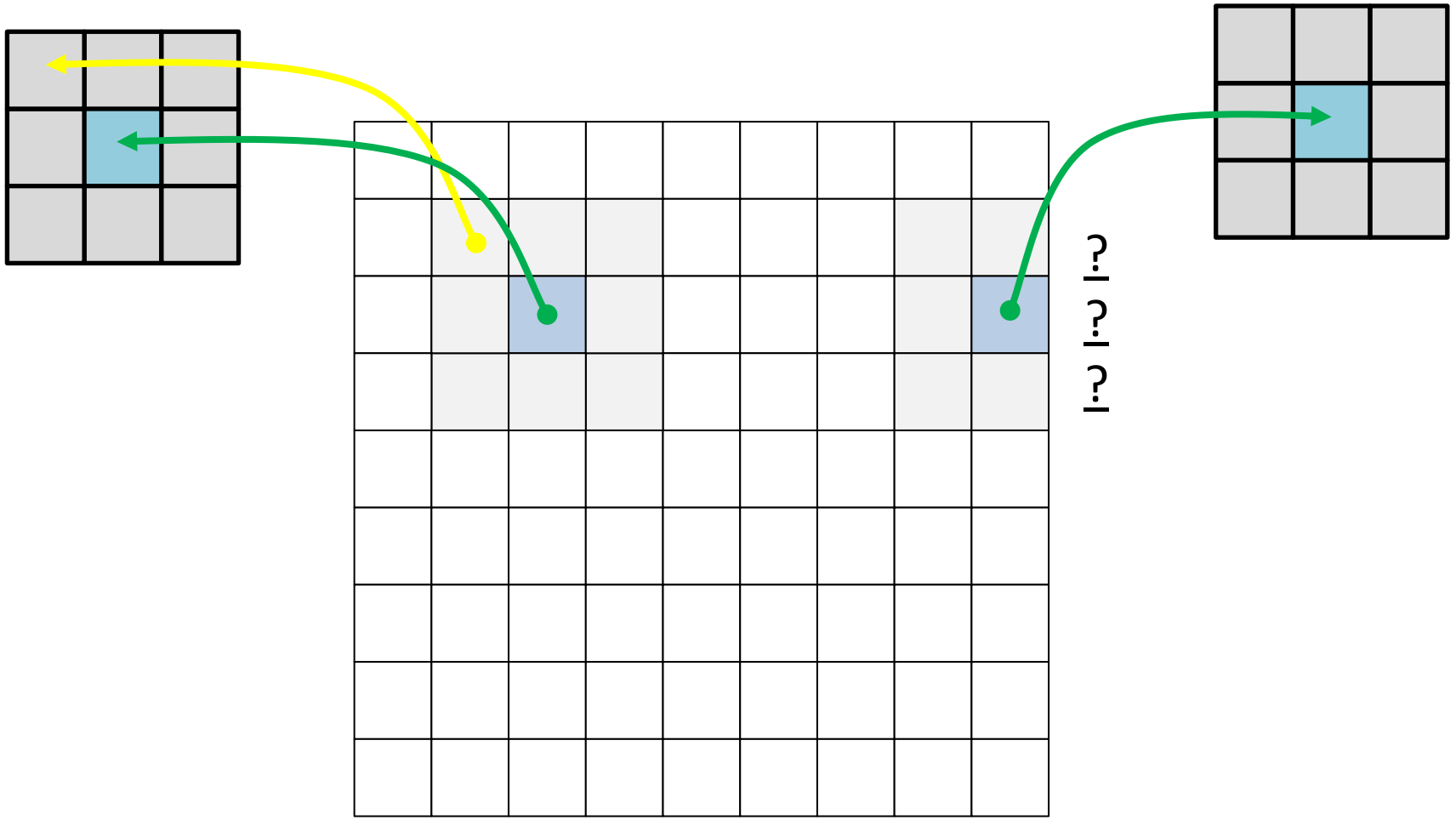
o

[ ]

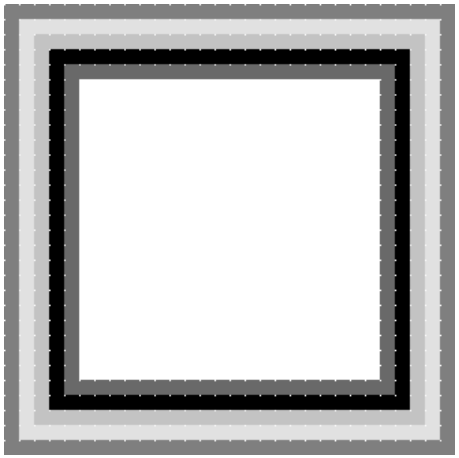
*

Sąsiedztwo - artefakty

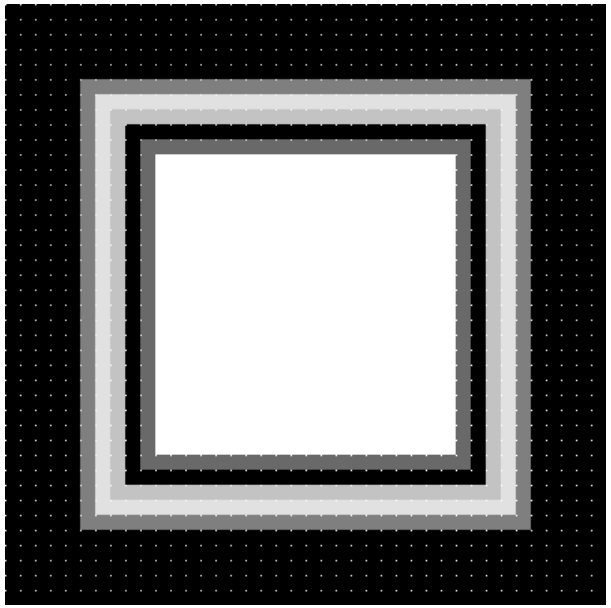
A co z marginesami?



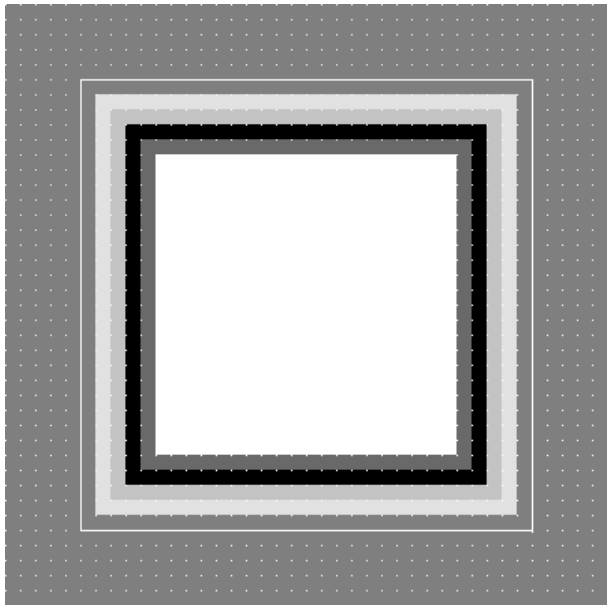
A co z marginesami?



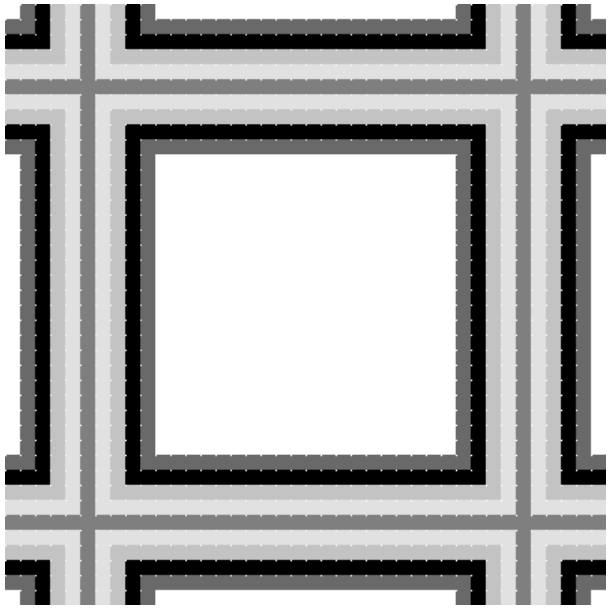
margines 5 pikseli



wypełnienie zerami

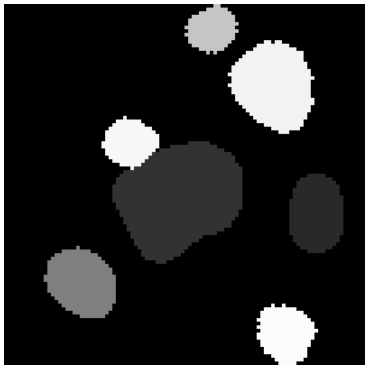


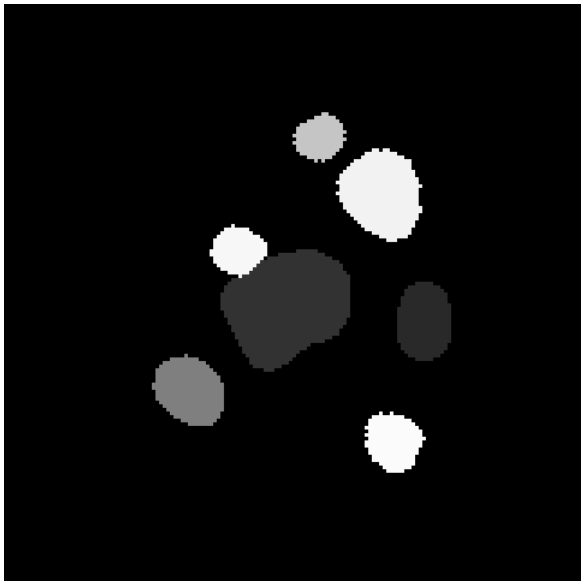
kopia brzegu



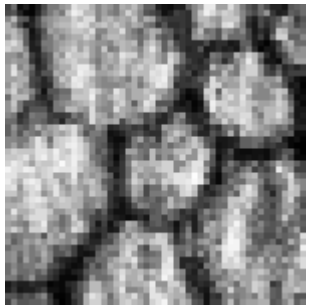
kopia lustrzana

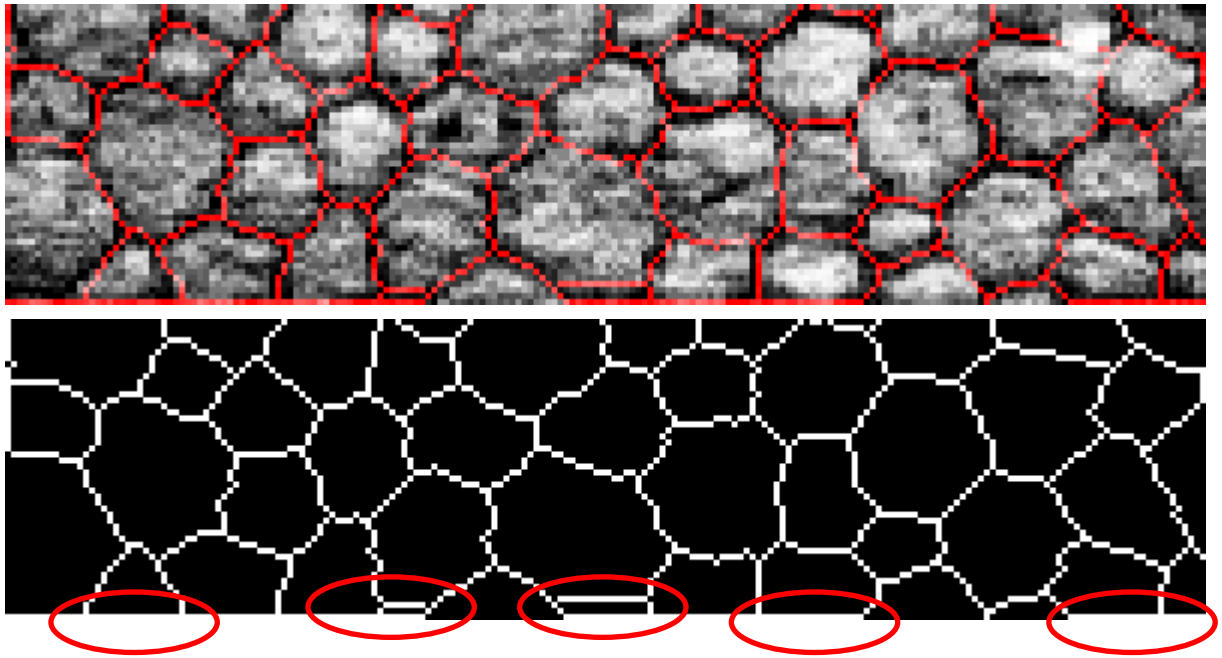
A co z marginesami?

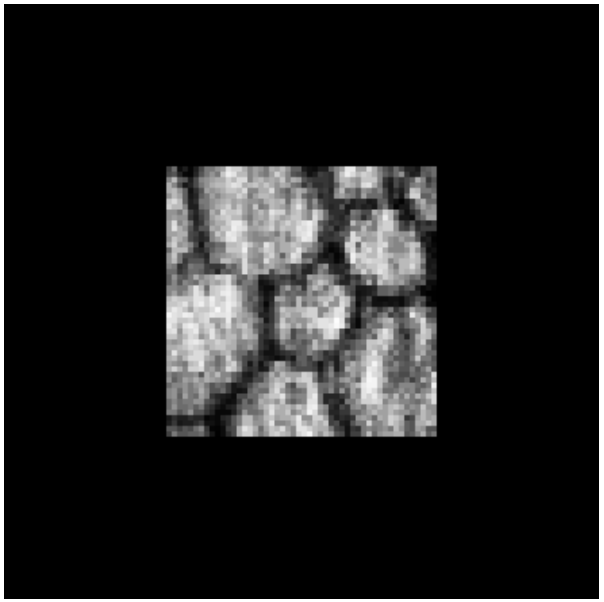


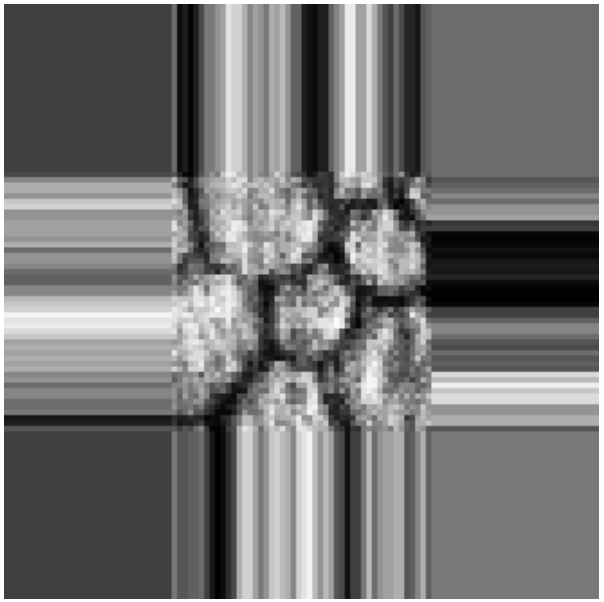


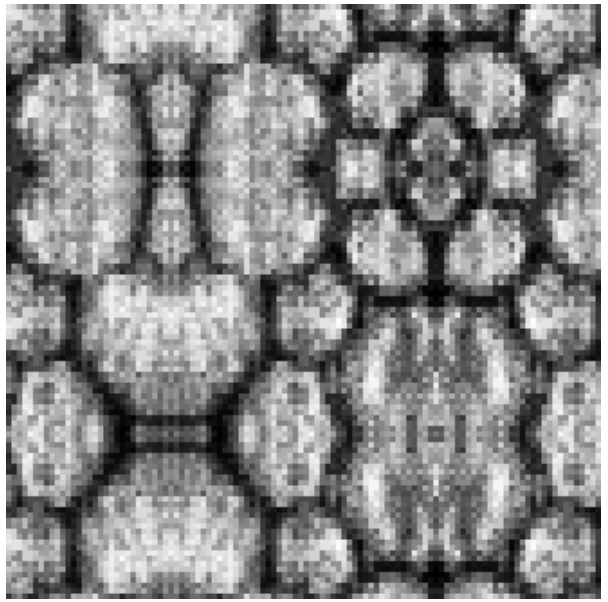
A co z marginesami?











```
// marginesy lustrzane, 2021.11.13, AP

Marg = 30;            // rozmiar marginesu

w = getWidth();
h = getHeight();

Marg = minOf(Marg, w-1);         // margines nie moze byc wiekszy niz rozmiar obrazu
Marg = minOf(Marg, h-1);         // margines nie moze byc wiekszy niz rozmiar obrazu

tytul =  File.getNameWithoutExtension(getTitle()) +"_marg_" + Marg;


// zaladowanie obrazu do tablicy
obraz = newArray(w*h);

for (y=0; y<h; y++)
          for (x=0; x<w; x++)
                    obraz[x + y *w] = getPixel(x, y);

// stworzenie nowego obrazu wiekszego o margines

obraz_nowy = newArray( (w + Marg * 2) * (h + Marg * 2) );
Array.fill(obraz_nowy,  0);

// kopiowanie czesci zasadniczej
for (i =  0; i < w; i++)
          for (j =  0; j < h; j++)
                    obraz_nowy[(i + Marg) + (w + Marg * 2) * (j + Marg)] =  obraz[i + w * j];


// ##############################################
// utworzenie nowego obrazka - marginesy wypelnione zerami
newImage(tytul + "_zero", "8-bit", w + Marg * 2, h + Marg * 2, 1);
for (y=0; y < h + Marg * 2; y++)
          for (x=0; x < w + Marg * 2; x++)
                    setPixel(x, y, obraz_nowy[x + y *(w + Marg * 2)] );
```

```
// marginesy lustrzane, 2021.11.13, AP     - c.d.

// ##############################################
// utworzenie nowego obrazka - marginesy wypelnione wartosciami brzegowymi

// kopiowanie marginesow
// gorny i dolny
for (i =  0; i < w; i++)
{
  for (k =  0; k <= Marg; k++)
  {
  obraz_nowy[i + Marg + (Marg - k)*(w + Marg * 2)] = obraz[i + w *  0];
  obraz_nowy[i + Marg + (h - 1 + Marg + k) * (w + Marg * 2)] = obraz[i + w* ( h - 1 )];
  }
}
// prawy i lewy
for (j =  0; j < h; j++)
{
  for (k =  0; k <= Marg; k++)
  {
  obraz_nowy[Marg - k + (j + Marg) *(w + Marg * 2)] = obraz[0*k + w * j];
  obraz_nowy[w - 1 + Marg + k + (j + Marg) *(w + Marg * 2)] = obraz[w - 1 -  0*k + w * j];
  }
}

// kopiowanie naroznikow
for (i =  0; i <= Marg; i++)
  for (j =  0; j <= Marg; j++)
  {
  obraz_nowy[Marg - i + (w + Marg * 2)*( Marg - j)] = obraz[0];
  obraz_nowy[w - 1 + Marg + i + (w + Marg * 2)*( Marg - j)] = obraz[w - 1];
  obraz_nowy[Marg - i + (w + Marg * 2)*(h - 1 + Marg + j)] = obraz[w*( h - 1)];
  obraz_nowy[w - 1 + Marg + i + (w + Marg * 2)*(h - 1 + Marg + j)] = obraz[w - 1 + w *( h - 1 )];
  }

newImage(tytul + "_brzeg", "8-bit", w + Marg * 2, h + Marg * 2, 1);
for (y=0; y < h + Marg * 2; y++)
           for (x=0; x < w + Marg * 2; x++)
                     setPixel(x, y, obraz_nowy[x + y *(w + Marg * 2)] );
```

```
// marginesy lustrzane, 2021.11.13, AP      - c.d. 2

// ##############################################
// utworzenie nowego obrazka - marginesy lustrzane

// kopiowanie marginesow
// gorny i dolny
for (i =  0; i < w; i++)
{
  for (k =  0; k <= Marg; k++)
  {
    obraz_nowy[i + Marg + (Marg - k)*(w + Marg * 2)] = obraz[i + w * k];
    obraz_nowy[i + Marg + (h - 1 + Marg + k) * (w + Marg * 2)] = obraz[i + w*( h - 1 - k)];
  }
}
// prawy i lewy
for (j =  0; j < h; j++)
{
  for (k =  0; k <= Marg; k++)
  {
    obraz_nowy[Marg - k + (j + Marg) *(w + Marg * 2)] = obraz[k + w * j];
    obraz_nowy[w - 1 + Marg + k + (j + Marg) *(w + Marg * 2)] = obraz[w - 1 - k + w * j];
  }
}

// kopiowanie naroznikow
for (i =  0; i <= Marg; i++)
  for (j =  0; j <= Marg; j++)
  {
    obraz_nowy[Marg - i + (w + Marg * 2)*( Marg - j)] = obraz[i + w * j];
    obraz_nowy[w - 1 + Marg + i + (w + Marg * 2)*( Marg - j)] = obraz[w - 1 - i + w * j];
    obraz_nowy[Marg - i + (w + Marg * 2)*(h - 1 + Marg + j)] = obraz[i + w*( h - 1 - j)];
    obraz_nowy[w - 1 + Marg + i + (w + Marg * 2)*(h - 1 + Marg + j)] = obraz[w - 1 - i + w *( h - 1 - j)];
  }

newImage(tytul + "_lustro", "8-bit", w + Marg * 2, h + Marg * 2, 1);
for (y=0; y < h + Marg * 2; y++)
    for (x=0; x < w + Marg * 2; x++)
        setPixel(x, y, obraz_nowy[x + y * (w + Marg * 2)] );
```

Filtry konwolucyjne (splotowe)

$$g(x) = (f \times h)(x) = \int_{-\infty}^{+\infty} f(x-t)h(t)dt$$

$$P'(m,n) = (w \times P)(m,n) = \sum_{i,j \in K} P(m-i, n-j)w(i,j)$$

$$w(i,j) = \begin{bmatrix} w_{-1,-1} & w_{0,-1} & w_{1,-1} \\ w_{-1,0} & w_{0,0} & w_{1,0} \\ w_{-1,1} & w_{0,1} & w_{1,1} \end{bmatrix}$$

Filtry są liniowe, gdy sa opisane funkcją:
- addytywną: $\varphi(f+g) = \varphi(f) + \varphi(g)$,
- jednorodną: $\varphi(\lambda f) = \lambda\, \varphi(f)$.

Filtry konwolucyjne (splotowe)

Kontrola zakresu wartości uzyskiwanych przy konwolucji

Dla współczynników nieujemnych: (co najmniej jeden większy od zera)

$$n = \frac{1}{\sum_{i,j \in K} w(i,j)}$$

$$P'(m,n) = \frac{1}{\sum_{i,j \in K} w(i,j)} \sum_{i,j \in K} P(m-i, n-j) w(i,j)$$

Dla współczynników dowolnych:

- skalowanie obrazu po konwolucji z zakresu (*min,max*) do (0,255) (normalizacja)
  - przesunięcie tła!
- ujemne - wyznaczenie wartości bezwzględnej z otrzymanego obrazu,
- limitowanie wartości wykraczających poza zakres (0,255)
  - niejawna binaryzacja

Filtry dolnoprzepustowe

0	1	0
1	1	1
0	1	0

1	1	1
1	1	1
1	1	1

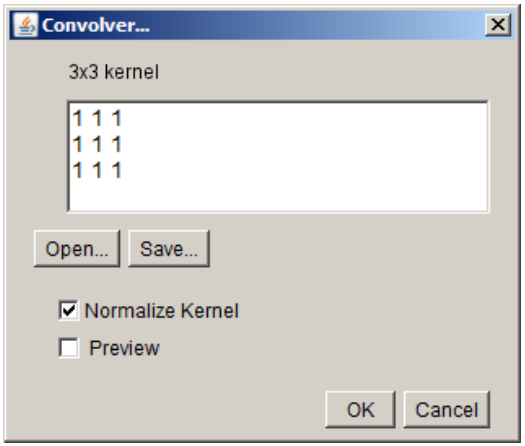
klasyczne filtry uśredniające

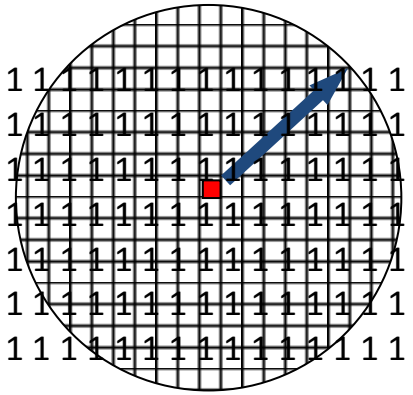
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

1	2	1
2	4	2
1	2	1

1	3	1
3	9	3
1	3	1

aproksymacja rozkładu Gaussa





Filtry dolnoprzepustowe:
+ redukują szum (wysokie częstotliwości),
+ wygładzają obraz,
- rozmywają krawędzie

Filtry górnoprzepustowe

- Gradienty (kierunkowe): (aproksymacja I pochodnej)
  - gradient Robertsa, maski Prewitta, Sobela,
- Laplasjany (bezkierunkowe) (suma drugich pochodnych cząstkowych)
- Maski wykrywające narożniki (i inne kształty)
  - maski Robinsona, Kircha, …

Gradient Robertsa

G_1

	1	0
	0	-1

G_2

	0	1
	-1	0

$$G = \sqrt{G_1^2 + G_2^2}$$

$$G = |G_1| + |G_2|$$

$$\Theta = \arctan(\frac{G_1}{G_2})$$

maski 2x2 nie oddają symetrycznego położenia punktu centralnego – powstaje przesunięcie w obrazie

Operatory Prewitta

1	**1**	0
1	0	**-1**
0	**-1**	**-1**

1	**1**	**1**
0	0	0
-1	**-1**	**-1**

0	**1**	**1**
-1	0	**1**
-1	**-1**	0

1	0	**-1**
1	0	**-1**
1	0	**-1**

-1	0	**1**
-1	0	**1**
-1	0	**1**

0	**-1**	**-1**
1	0	**-1**
1	**1**	0

-1	**-1**	**-1**
0	0	0
1	**1**	**1**

-1	**-1**	0
-1	0	**1**
0	**1**	**1**

Operatory Prewitta

-1	**-1**	0
-1	0	**1**
0	**1**	**1**

-1	**-1**	**-1**
0	0	0
1	**1**	**1**

0	**-1**	**-1**
1	0	**-1**
1	**1**	0

-1	0	**1**
-1	0	**1**
-1	0	**1**

1	0	**-1**
1	0	**-1**
1	0	**-1**

0	**1**	**1**
-1	0	**1**
-1	**-1**	0

1	**1**	**1**
0	0	0
-1	**-1**	**-1**

1	**1**	0
1	0	**-1**
0	**-1**	**-1**

jasny obiekt – ciemne tło

Operatory Sobela

2	**1**	0
1	0	**-1**
0	**-1**	**-2**

1	**2**	**1**
0	0	0
-1	**-2**	**-1**

0	**1**	**2**
-1	0	**1**
-2	**-1**	0

1	0	**-1**
2	0	**-2**
1	0	**-1**

-1	0	**1**
-2	0	**2**
-1	0	**1**

0	**-1**	**-2**
1	0	**-1**
2	**1**	0

-1	**-2**	**-1**
0	0	0
1	**2**	**1**

-2	**-1**	0
-1	0	**1**
0	**1**	**2**

Maski Robinsona (wykrywanie narożników)

1	1	1
1	-2	-1
1	-1	-2

1	1	1
1	-2	1
-1	-2	-1

1	1	1
-1	-2	1
-2	-1	1

1	1	-1
1	-2	-2
1	1	-1

-1	1	1
-2	-2	1
-1	1	1

1	-1	-2
1	-2	-1
1	1	1

-1	-2	-1
1	-2	1
1	1	1

-2	-1	1
-1	-2	1
1	1	1

Maski Kircha (wykrywanie narożników)

3	3	3
3	0	-5
3	-5	-5

3	3	3
3	0	3
-5	-5	-5

3	3	3
-5	0	3
-5	-5	3

3	3	-5
3	0	-5
3	3	-5

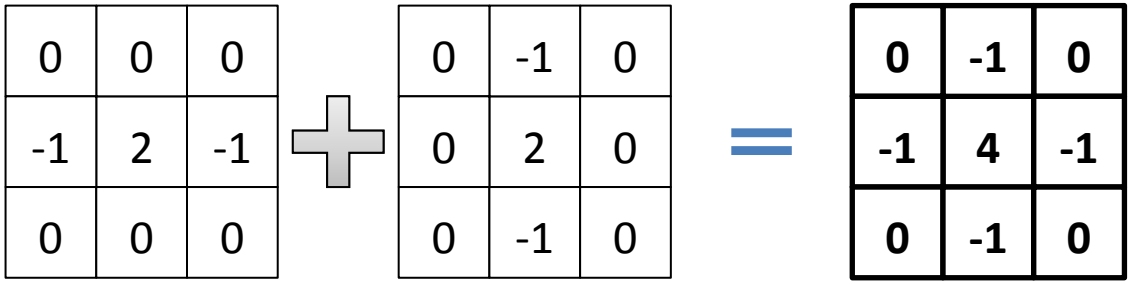
-5	3	3
-5	0	3
-5	3	3

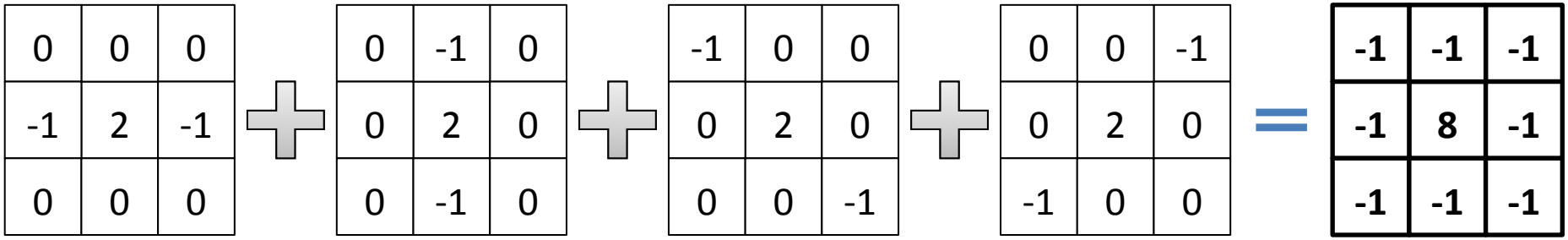
3	-5	-5
3	0	-5
3	3	3

-5	-5	-5
3	0	3
3	3	3

-5	-5	3
-5	0	3
3	3	3

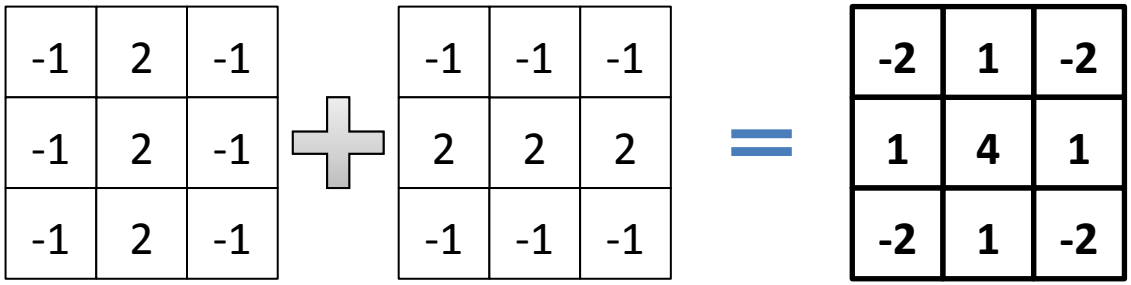
Operatory Laplace’a

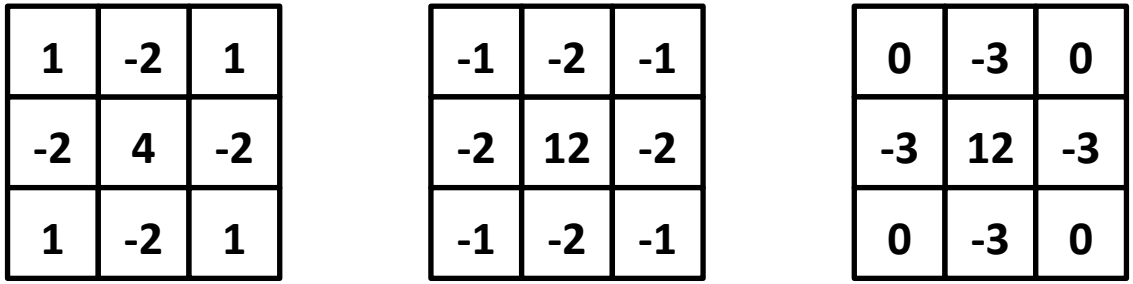




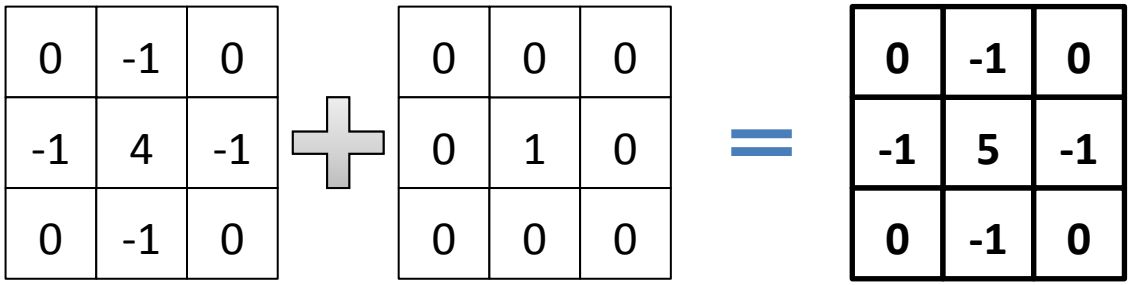
Kierunkowe pochodne cząstkowe

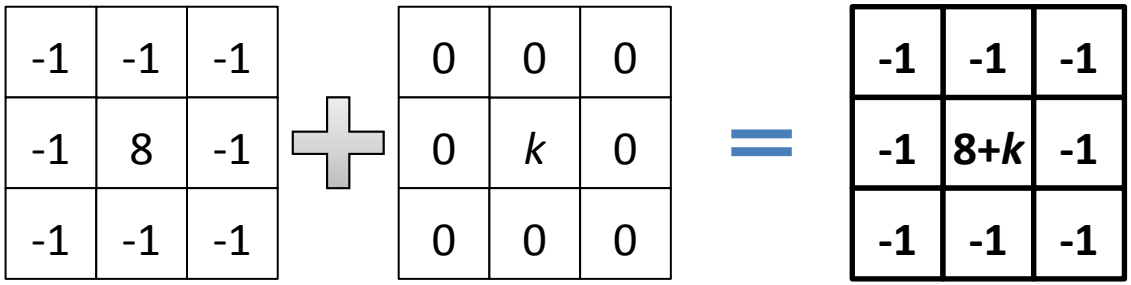
Operatory Laplace’a





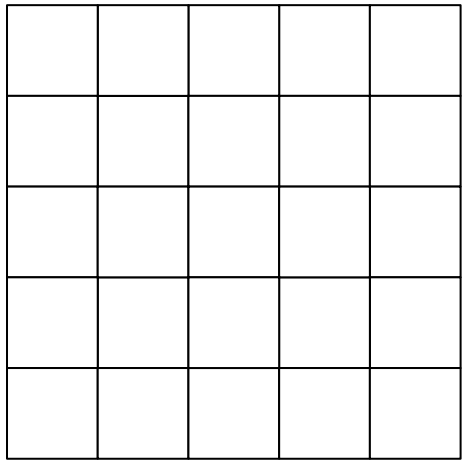
Operatory Laplace'a - wyostrzanie

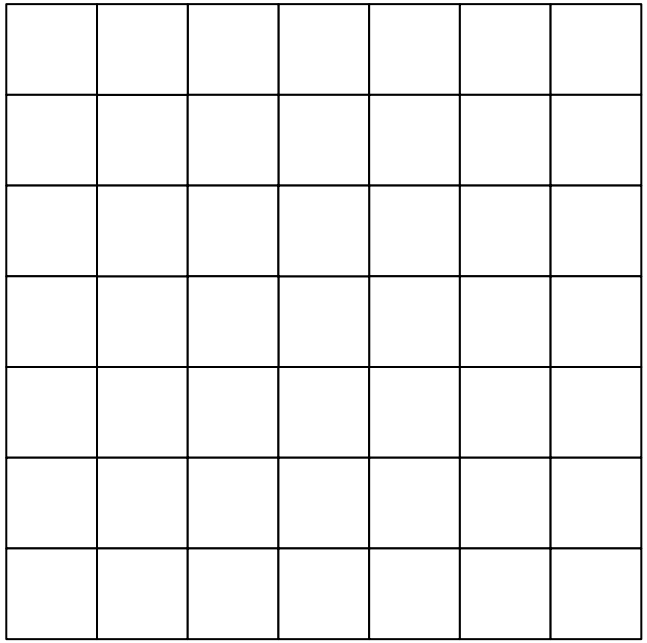


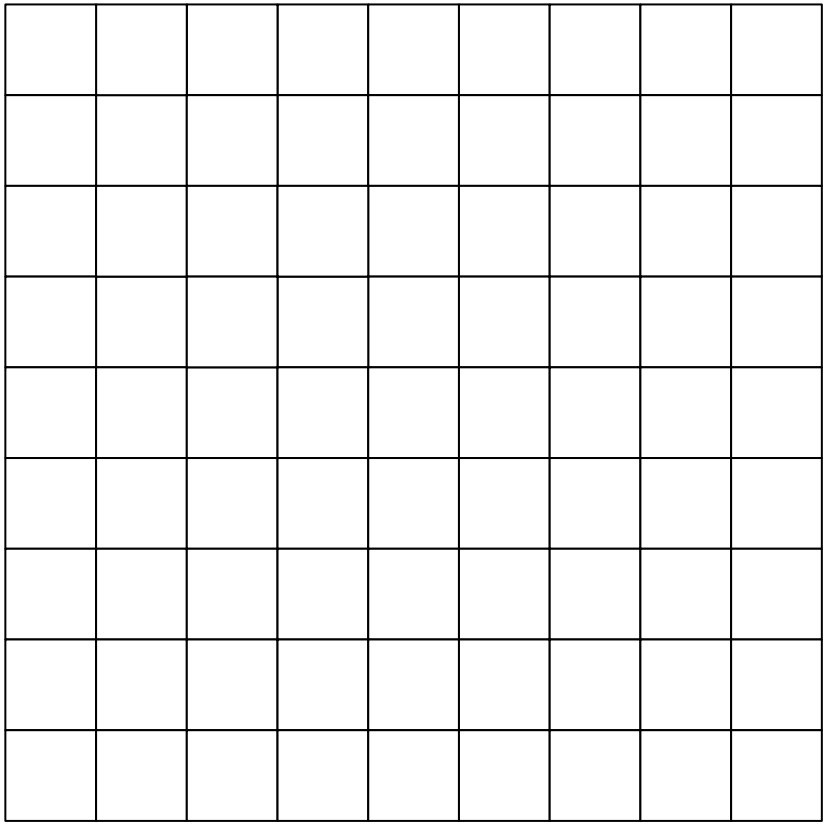


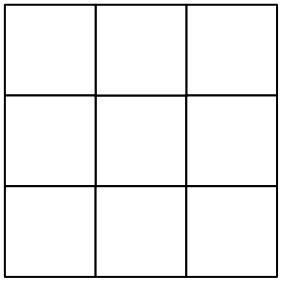
k=1,2,3, …

W TYM MIEJSCU PRZEWIDZIANY JEST POKAZ NA ŻYWO (19.11.2021) …..









Filtr Gaussa

1D

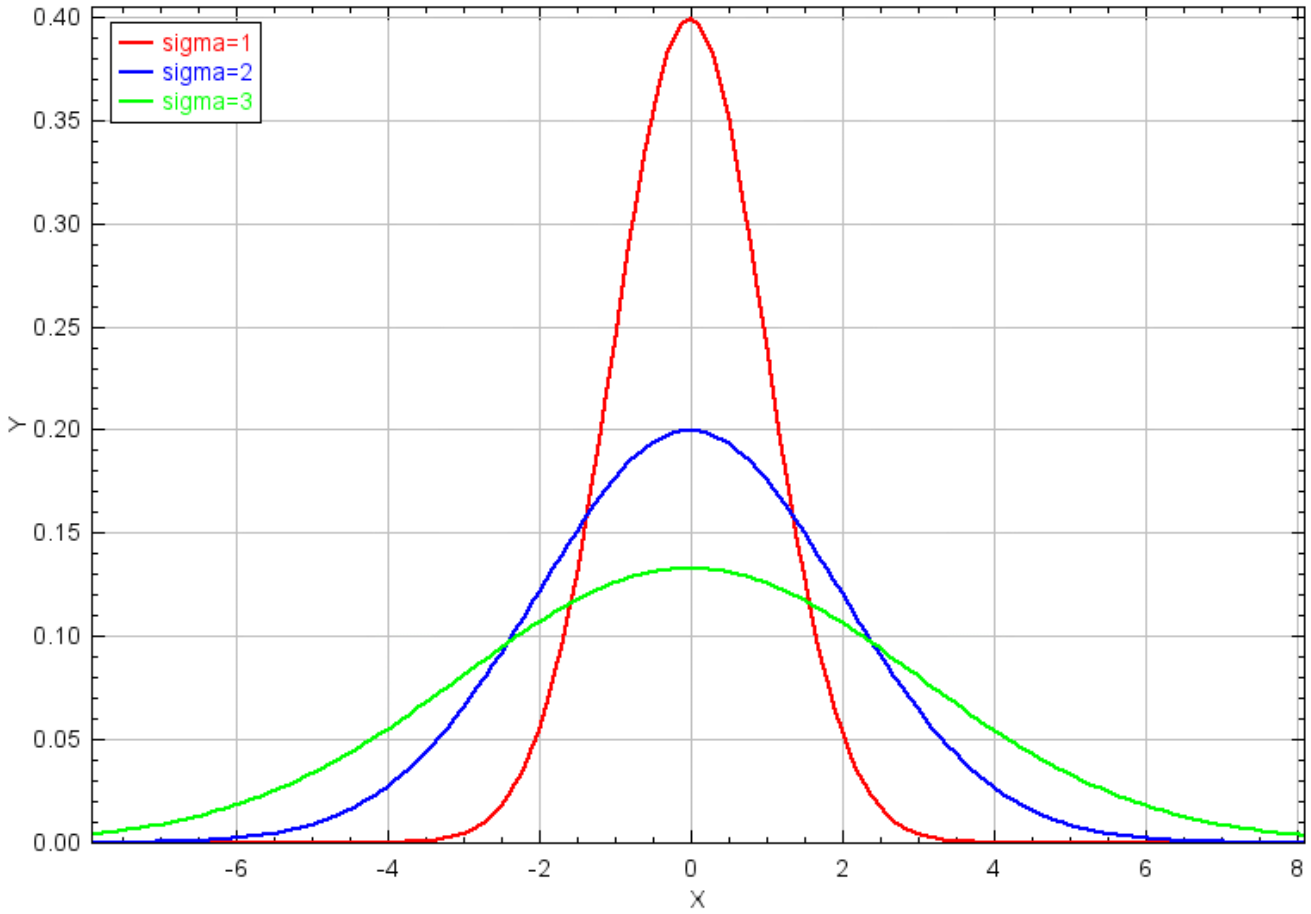
$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

```
s = 1.0;

N = 8; // rozmiar

dokladnosc = 0.1;

tab_x = newArray((N*2)/dokladnosc);
tab_y = newArray((N*2)/dokladnosc);

Plot.create("Gauss", "X", "Y");

for(i =-N; i<N+dokladnosc; i+=dokladnosc)
{
          tab_x[(i+N)/dokladnosc] = i;
          tab_y[(i+N)/dokladnosc] =  Gauss1D(i, s);
}

function Gauss(x, y, sigma)
{
          return  (1.0 / (sqrt(2*PI* sigma*sigma))) * exp( -(x*x )/(2.0*sigma*sigma) ) ;
}

Plot.add("line", tab_x, tab_y);
```

Filtr Gaussa

1D

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$



Filtr Gaussa

2D

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$ 1D

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$ 2D

```
s = 3.0;

N = 4; // rozmiar 2*N+1, czyli 9

for(i=-N; i<=N; i++)
{
        linia="";

        for(j=-N; j<=N; j++)
                linia+= " " + round( 100* Gauss(i, j, s));
        print(linia);
}


function Gauss(x, y, sigma)
{
        return  (1.0 / (2*PI* sigma*sigma)) * exp( -(x*x + y*y)/(2.0*sigma*sigma) ) ;
}
```

Filtr Gaussa

sigma=1

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	1	0	0	0
0	0	1	6	10	6	1	0	0
0	0	2	10	16	10	2	0	0
0	0	1	6	10	6	1	0	0
0	0	0	1	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

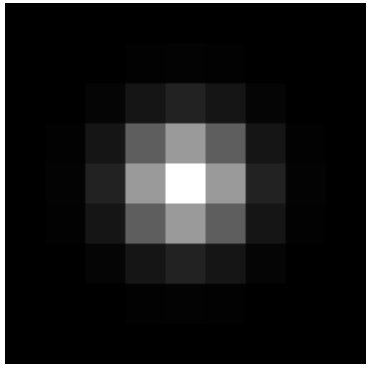
sigma=2

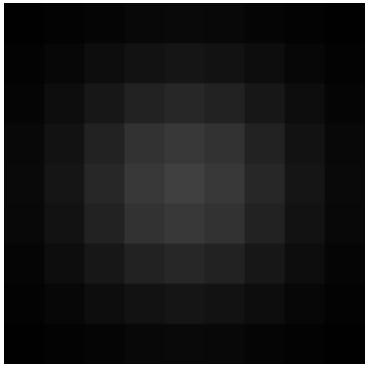
0	0	0	0	1	0	0	0	0
0	0	1	1	1	1	1	0	0
0	1	1	2	2	2	1	1	0
0	1	2	3	4	3	2	1	0
1	1	2	4	4	4	2	1	1
0	1	2	3	4	3	2	1	0
0	1	1	2	2	2	1	1	0
0	0	1	1	1	1	1	0	0
0	0	0	0	1	0	0	0	0

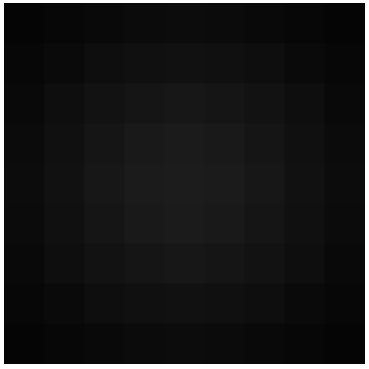
sigma=3

0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1
1	1	1	2	2	2	1	1	1
1	1	1	2	2	2	1	1	1
1	1	1	2	2	2	1	1	1
1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	0	0

kernel:







Laplacian of Gaussian

Złożenie transformacji 1) Gaussian, 2) Laplacian

$$\mathrm{LoG}(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

```
s = 3.0;

N = 4; // rozmiar 2*N+1, czyli 9

for(i=-N; i<=N; i++)
{
          linia="";

          for(j=-N; j<=N; j++)
                    linia+= " " + round( 100* LoG(i, j, s));
          print(linia);
}


function LoG(x, y, sg)
{
   return - (1.0 / (PI* sg*sg*sg*sg)) * (1.0 - (x*x + y*y)/(2.0*sg*sg)) * exp( -(x*x + y*y)/(2.0*sg*sg) ) ;
}
```

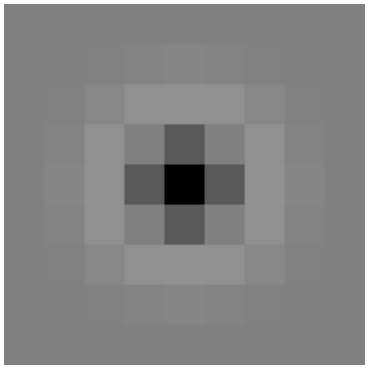
Laplacian of Gaussian

sigma=1.0

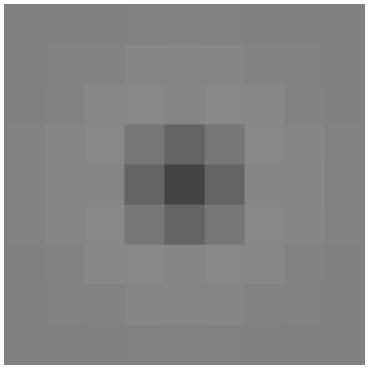
0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	2	4	4	4	2	0	0
0	1	4	0	-10	0	4	1	0
0	1	4	-10	-32	-10	4	1	0
0	1	4	0	-10	0	4	1	0
0	0	2	4	4	4	2	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0

sigma=1.2

0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0
0	1	2	2	1	2	2	1	0
0	1	2	-2	-7	-2	2	1	0
0	1	1	-7	**-15**	-7	1	1	0
0	1	2	-2	-7	-2	2	1	0
0	1	2	2	1	2	2	1	0
0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0



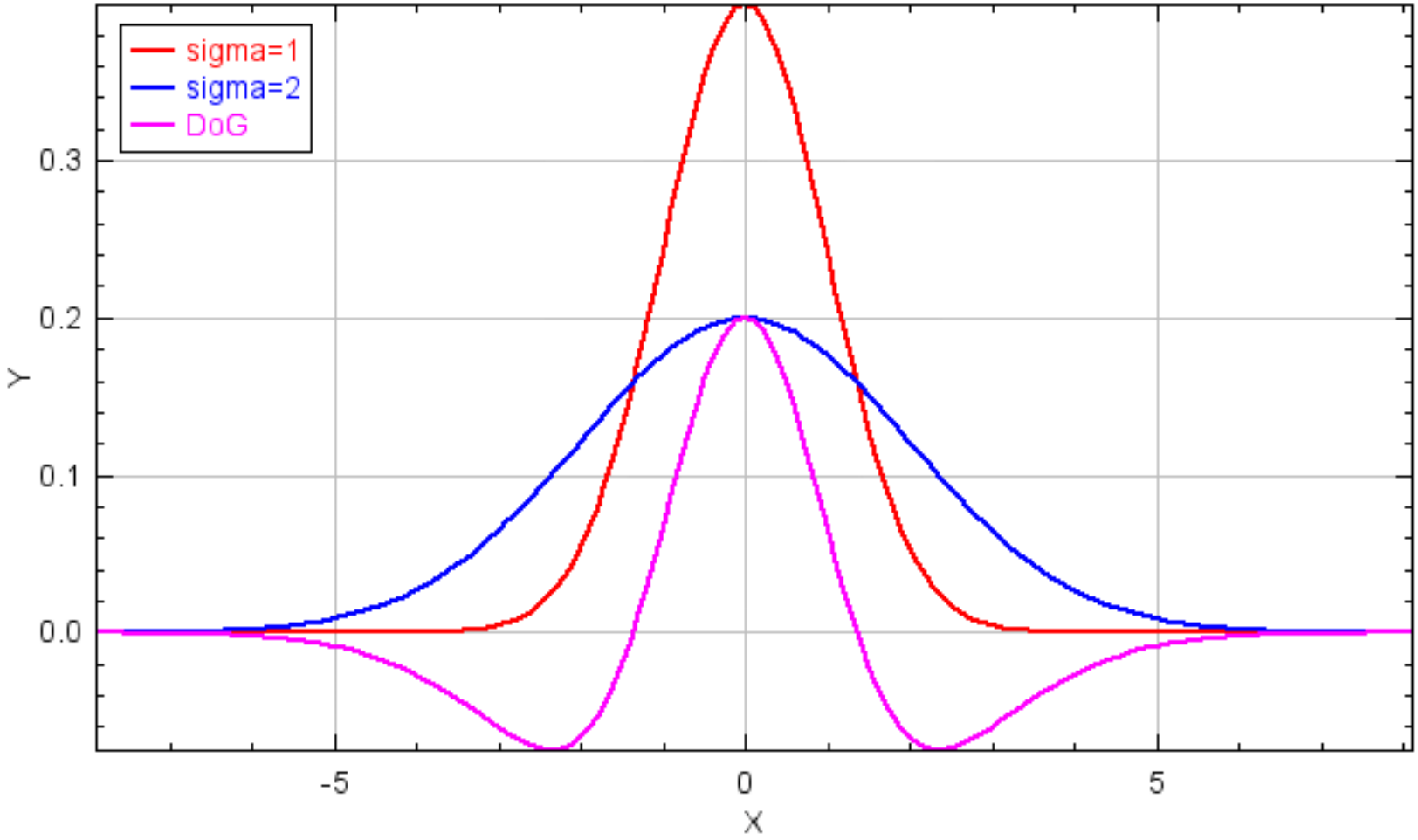
obrazy przesunięte +128



BLOB DETECTION

Mexican Hat

Difference of Gaussian



Difference of Gaussian

sigma=1

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	1	0	0	0
0	0	1	6	10	6	1	0	0
0	0	2	10	16	10	2	0	0
0	0	1	6	10	6	1	0	0
0	0	0	1	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

sigma=2

0	0	0	0	1	0	0	0	0
0	0	1	1	1	1	1	0	0
0	1	1	2	2	2	1	1	0
0	1	2	3	4	3	2	1	0
1	1	2	4	4	4	2	1	1
0	1	2	3	4	3	2	1	0
0	1	1	2	2	2	1	1	0
0	0	1	1	1	1	1	0	0
0	0	0	0	1	0	0	0	0

sigma=3

0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1
1	1	1	2	2	2	1	1	1
1	1	1	2	2	2	1	1	1
1	1	1	2	2	2	1	1	1
1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	0	0

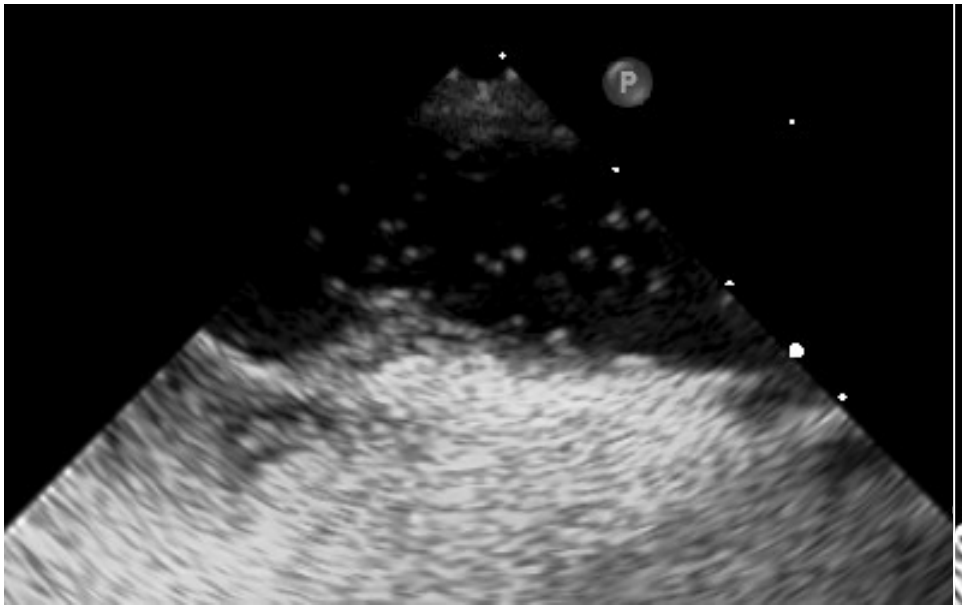
sigma=1 - sigma=2

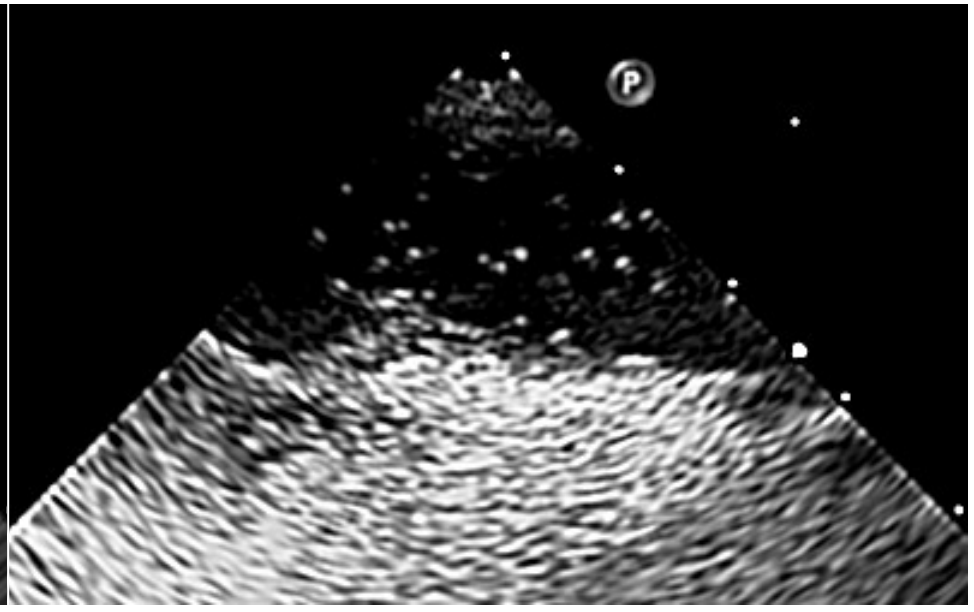
0	0	0	0	-1	0	0	0	0
0	0	-1	-1	-1	-1	-1	0	0
0	-1	-1	-1	0	-1	-1	-1	0
0	-1	-1	3	6	3	-1	-1	0
-1	-1	0	6	12	6	0	-1	-1
0	-1	-1	3	6	3	-1	-1	0
0	-1	-1	-1	0	-1	-1	-1	0
0	0	-1	-1	-1	-1	-1	0	0
0	0	0	0	-1	0	0	0	0

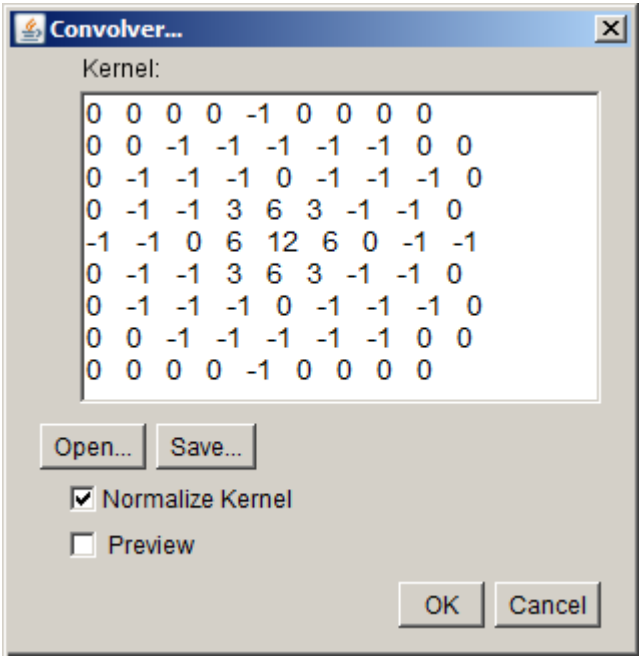
sigma=1 - sigma=3

0	0	-1	-1	-1	-1	-1	0	0
0	-1	-1	-1	-1	-1	-1	-1	0
-1	-1	-1	0	1	0	-1	-1	-1
-1	-1	0	4	8	4	0	-1	-1
-1	-1	1	8	14	8	1	-1	-1
-1	-1	0	4	8	4	0	-1	-1
-1	-1	-1	0	1	0	-1	-1	-1
0	-1	-1	-1	-1	-1	-1	-1	0
0	0	-1	-1	-1	-1	-1	0	0

Difference of Gaussian

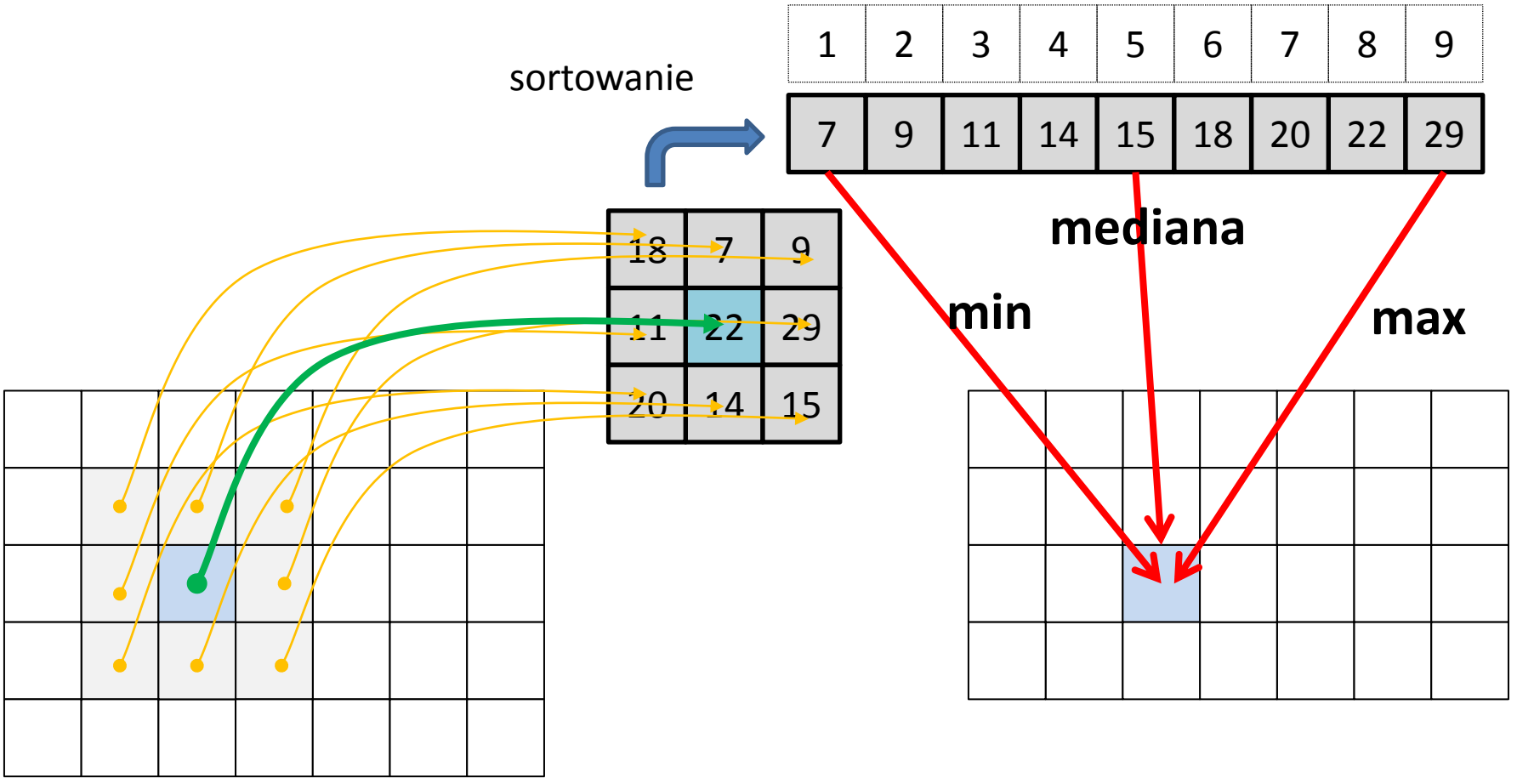






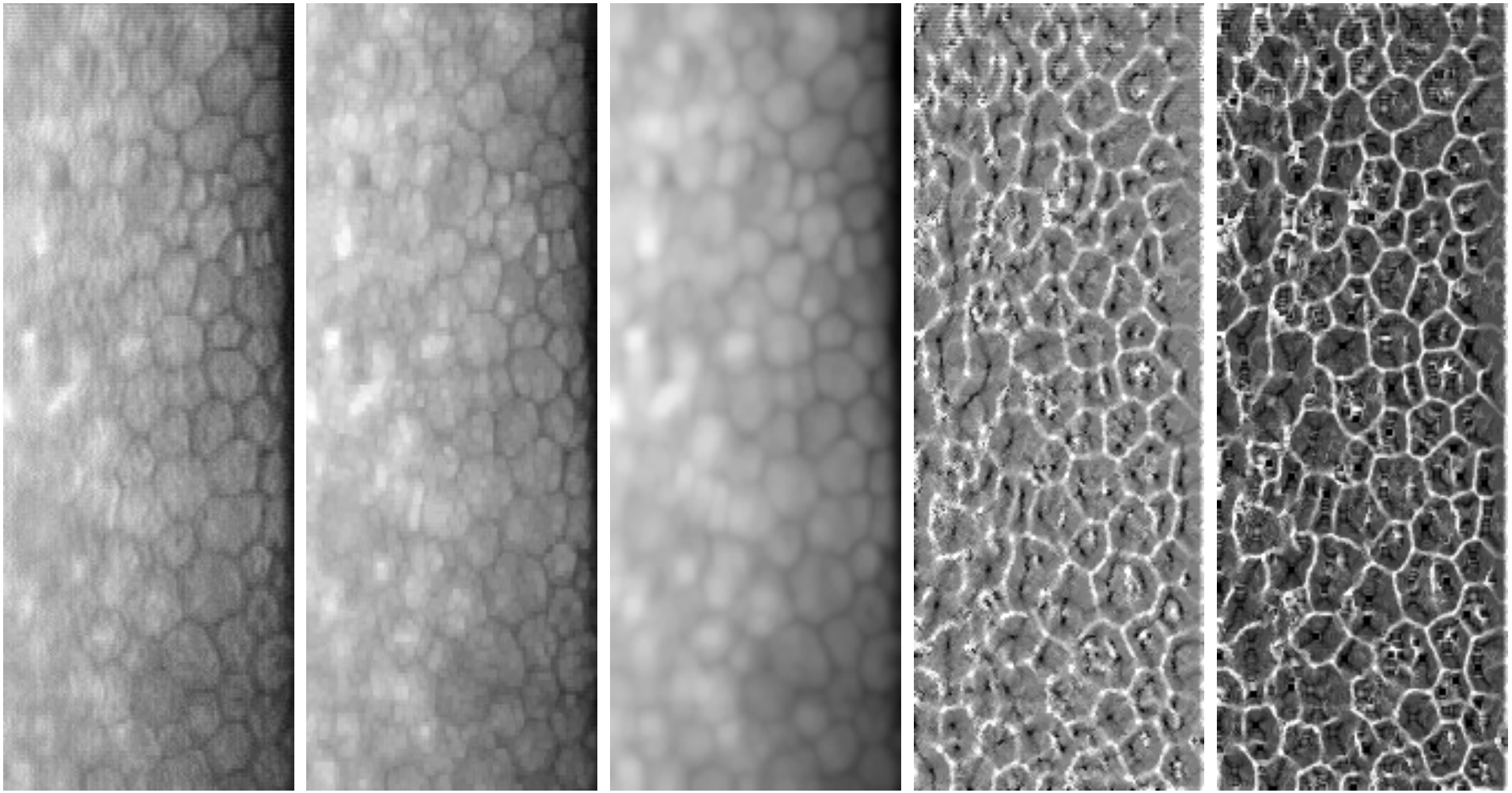
Sorysz, J., Sorysz, D., Piórkowski, A.: Segmentation of a First Generation Agent Bubbles in the B-Mode Echocardiographic Images. IEEE EMBS ICS 2020. Springer 2022, AISC, vol 1360, pp. 127-135

Filtry rankingowe



- nie wprowadzają nowych wartości poziomów jasności
- mediana w dużym stopniu pozostawia definicje krawędzi

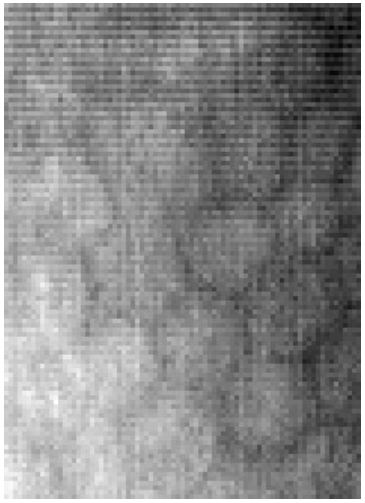
Filtr maksymalny – przykład wykorzystania



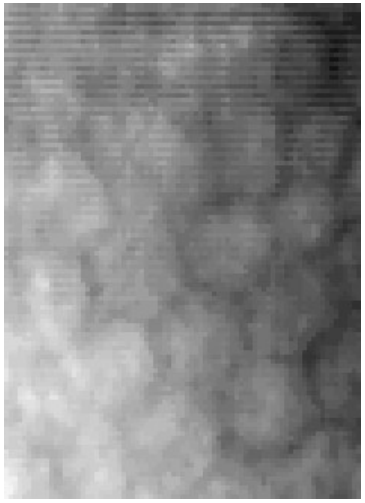
org | **max** [3x3] | mean r=3.5 **max** [3x3] | mean r=3.5 SDAr3 | mean r=3.5 **max** [3x3] SDAr3

Piorkowski A.: A Statistical Dominance Algorithm for Edge Detection and Segmentation of Medical Images. AISC vol. 471, Springer 2016, pp. 3-14.

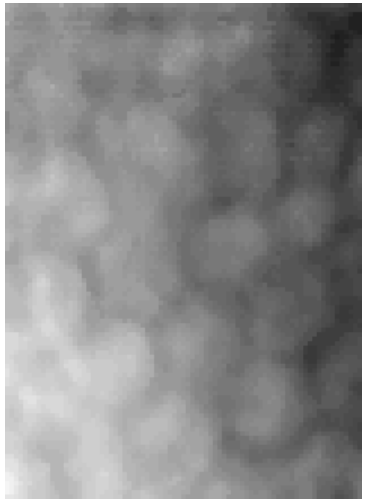
Mediana a uśrednianie



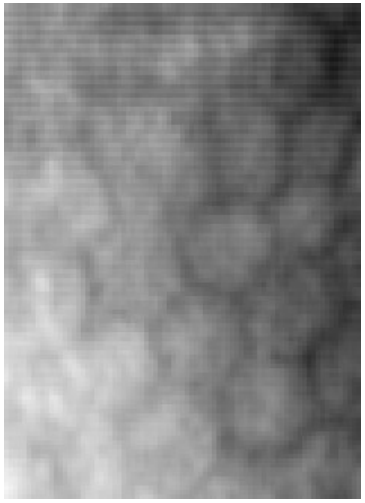
obraz źródłowy



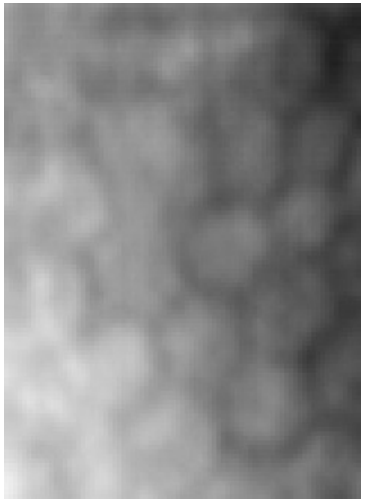
mediana r=1



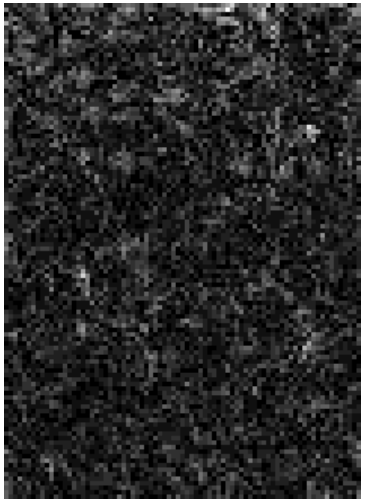
mediana r=2

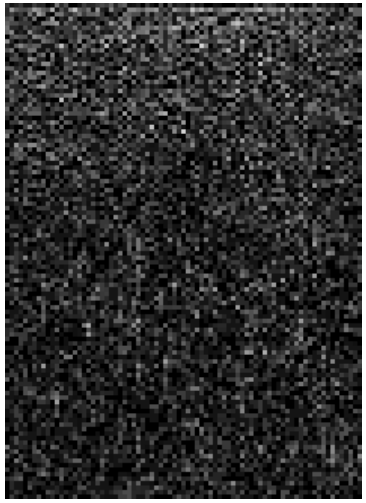


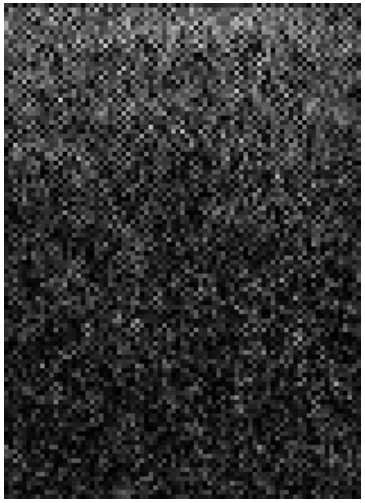
średnia r=1

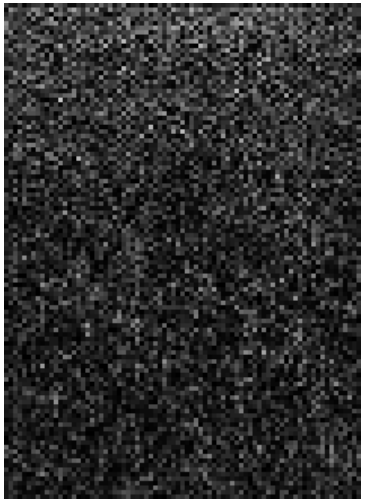


średnia r=2





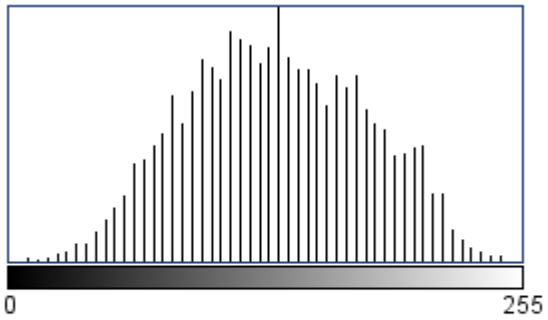




Różnica między medianą r=2
a średnią r=2, pomnożona 8x

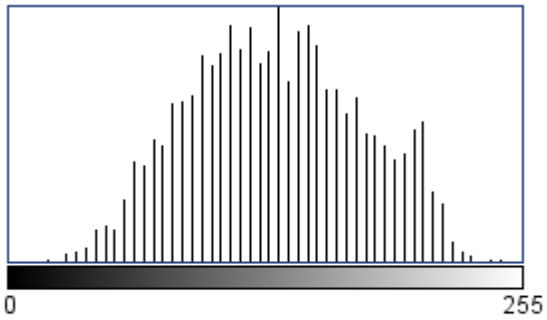
Różnica w stosunku do obrazu źródłowego, pomnożona x4

Mediana a uśrednianie



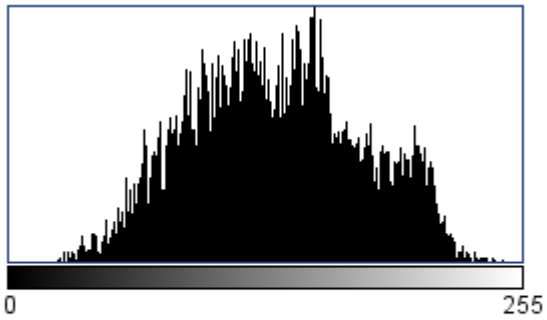
N: 8690 Min: 0
Mean: 132.410 Max: 255
StdDev: 45.093 Mode: 134 (390)
Value: 232 Count: 0

obraz źródłowy



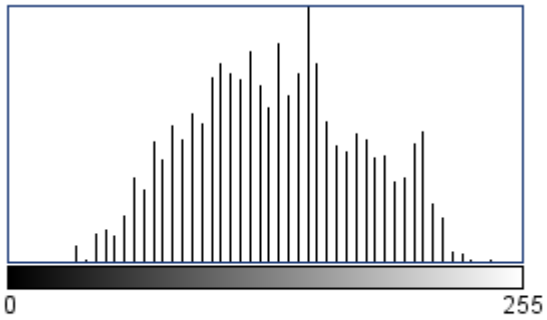
N: 8690 Min: 19
Mean: 132.444 Max: 245
StdDev: 43.139 Mode: 134 (393)
Value: 13 Count: 0

mediana r=1



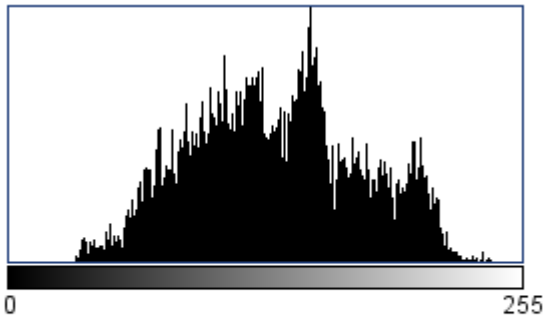
N: 8690 Min: 24
Mean: 132.412 Max: 246
StdDev: 42.930 Mode: 152 (94)

średnia r=1



N: 8690 Min: 28
Mean: 132.548 Max: 245
StdDev: 42.486 Mode: 149 (464)
Value: 9 Count: 0

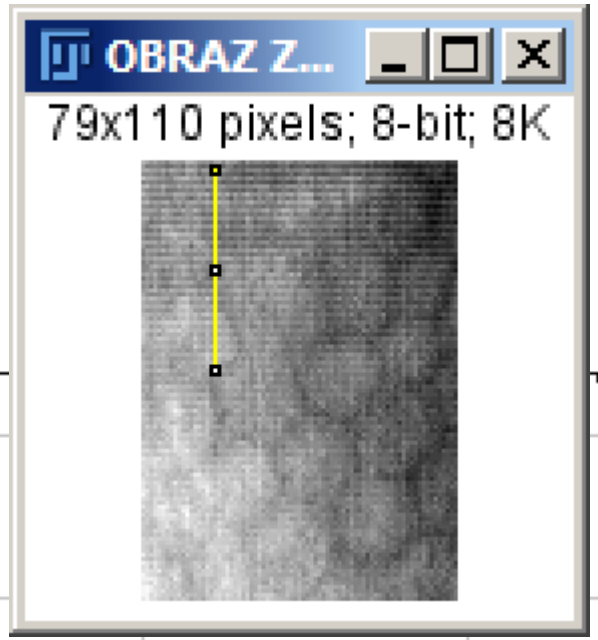
mediana r=2

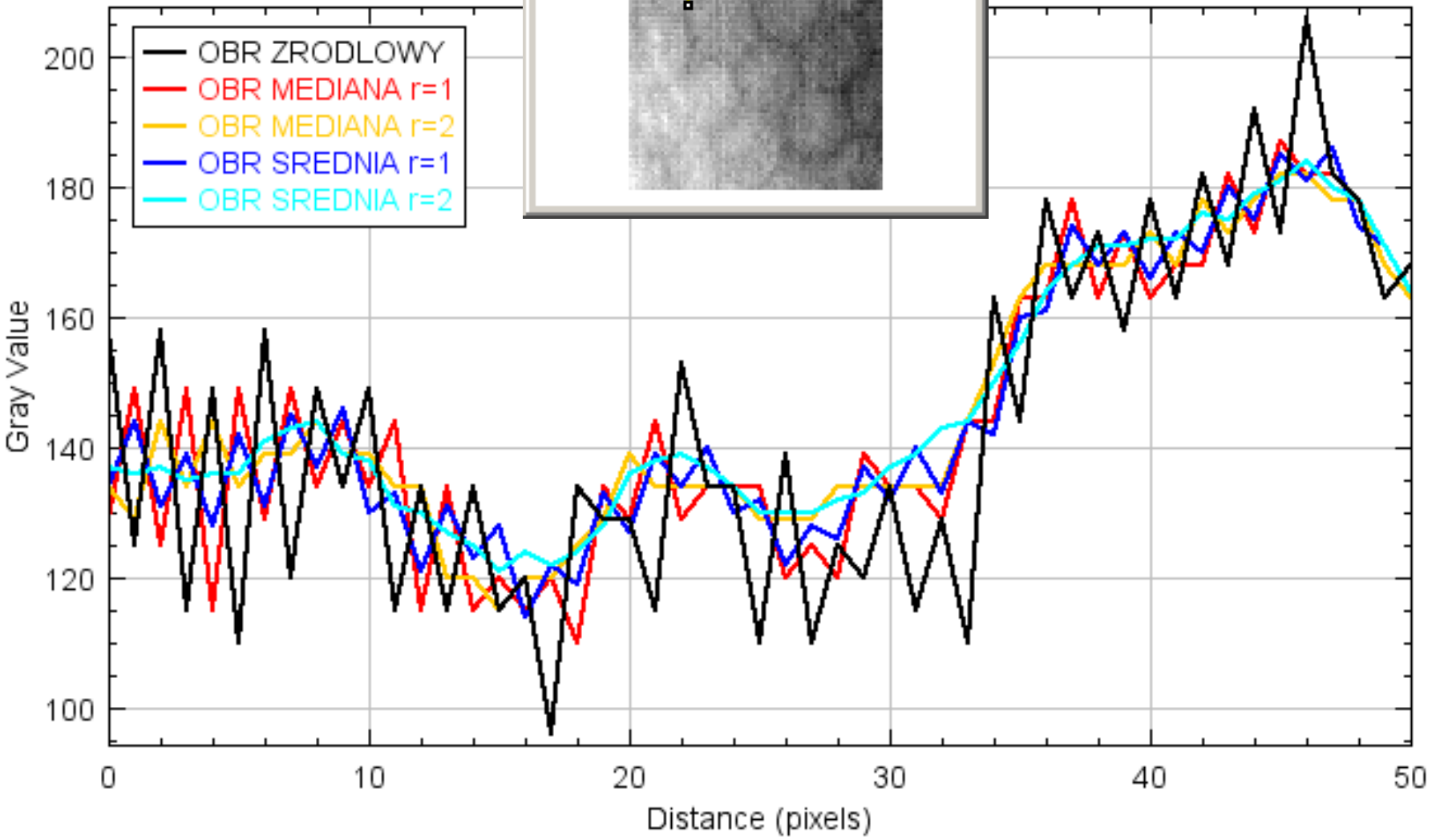


N: 8690 Min: 33
Mean: 132.461 Max: 240
StdDev: 42.432 Mode: 150 (114)
Value: --- Count: ---

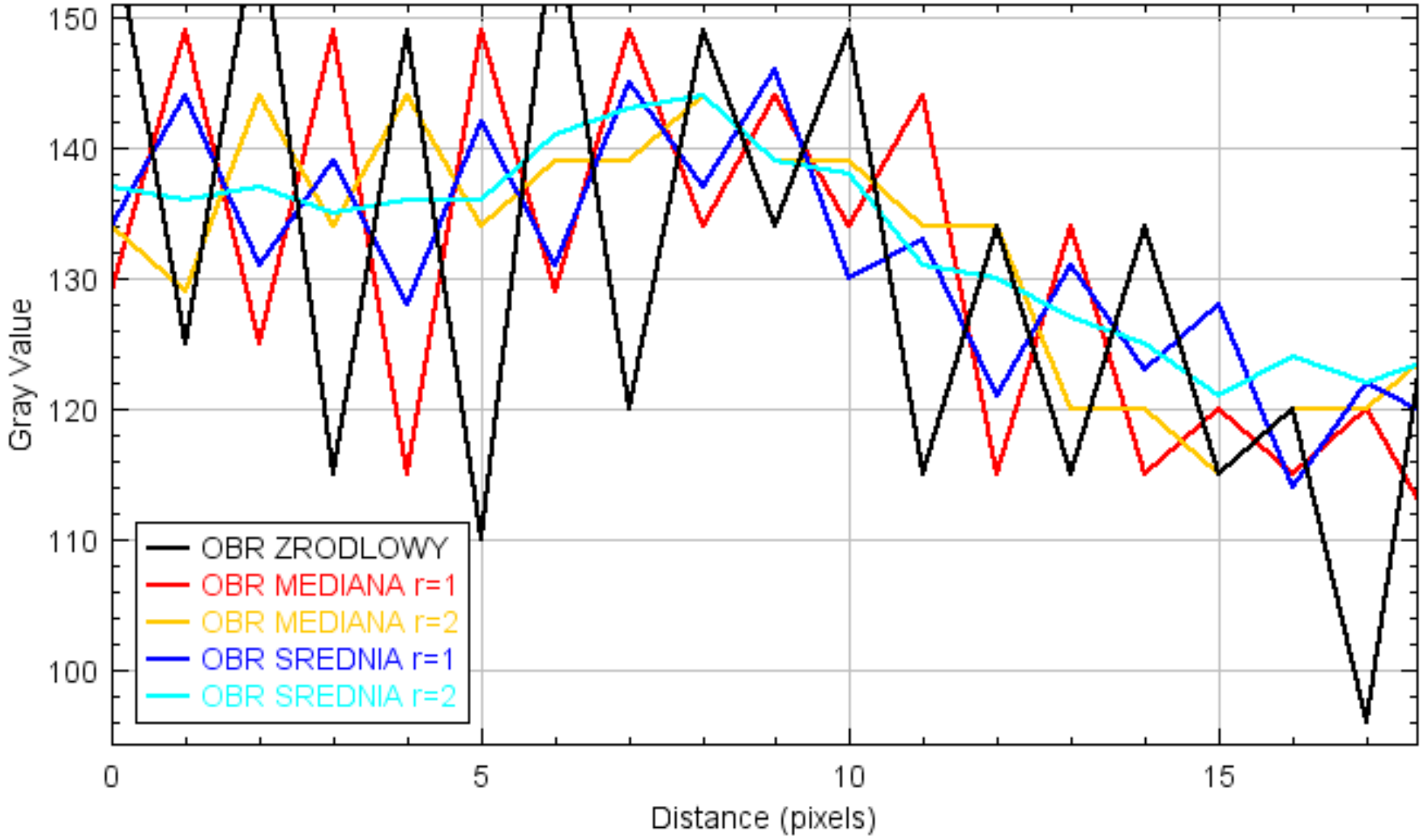
średnia r=2

Mediana a uśrednianie

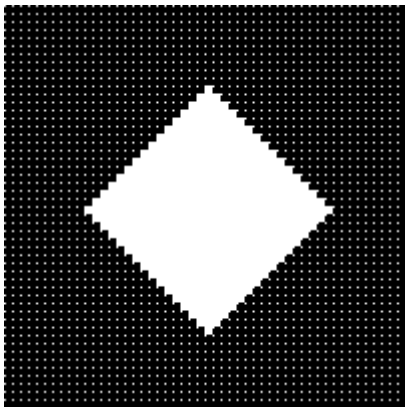


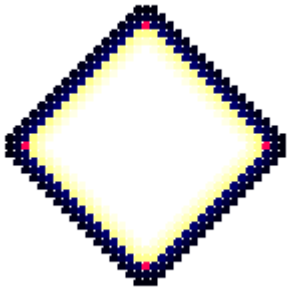


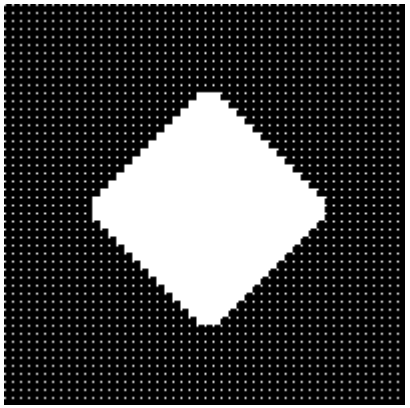
Mediana a uśrednianie



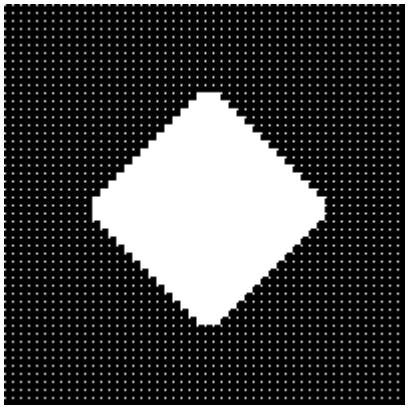
Mediana a uśrednianie



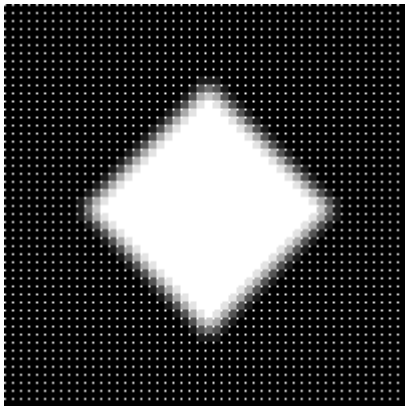




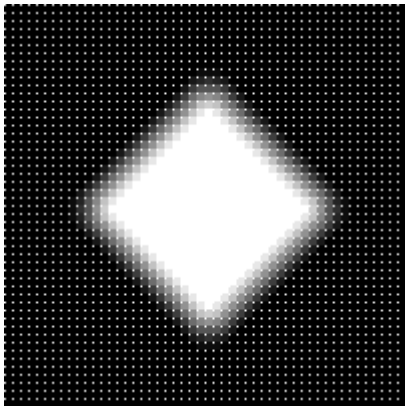
mediana r=1



mediana r=2

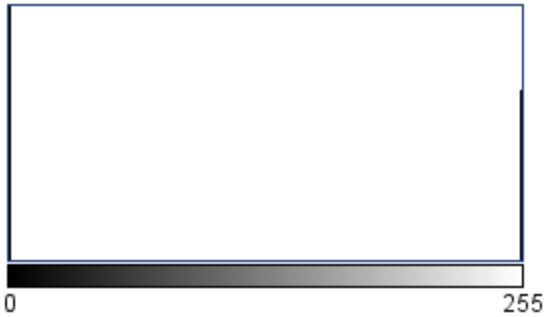


średnia r=1



średnia r=2

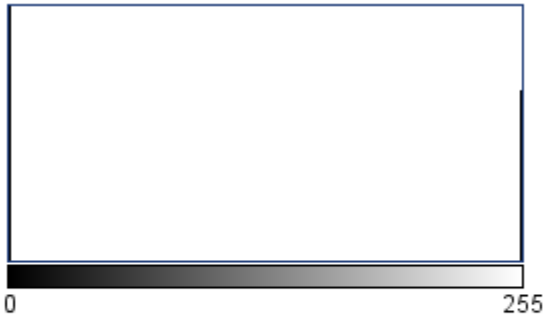
Mediana a uśrednianie



N: 2500
Mean: 49.062
StdDev: 100.537
Value: 189

Min: 0
Max: 255
Mode: 0 (2019)
Count: 0

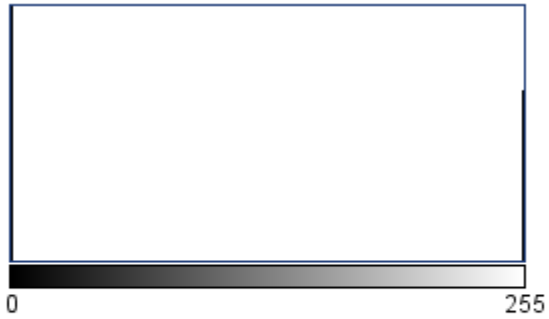
mediana r=1



N: 2500
Mean: 48.654
StdDev: 100.218
Value: 254

Min: 0
Max: 255
Mode: 0 (2023)
Count: 0

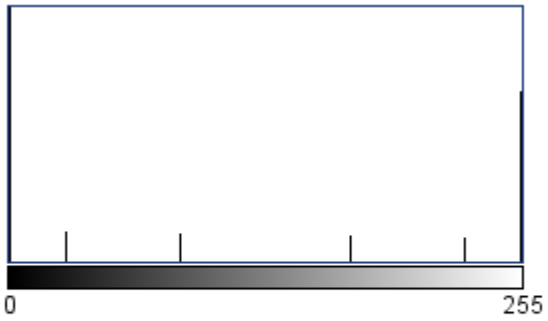
mediana r=2



N: 2500
Mean: 48.654
StdDev: 100.218
Value: 241

Min: 0
Max: 255
Mode: 0 (2023)
Count: 0

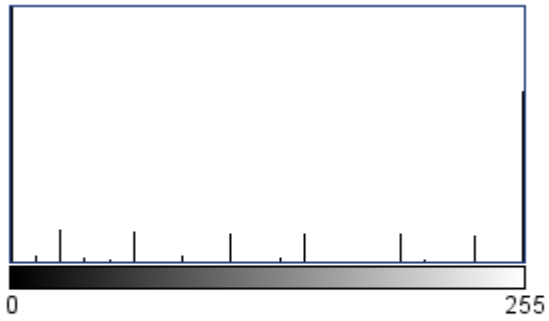
średnia r=1



N: 2500
Mean: 49.059
StdDev: 95.329

Min: 0
Max: 255
Mode: 0 (1891)

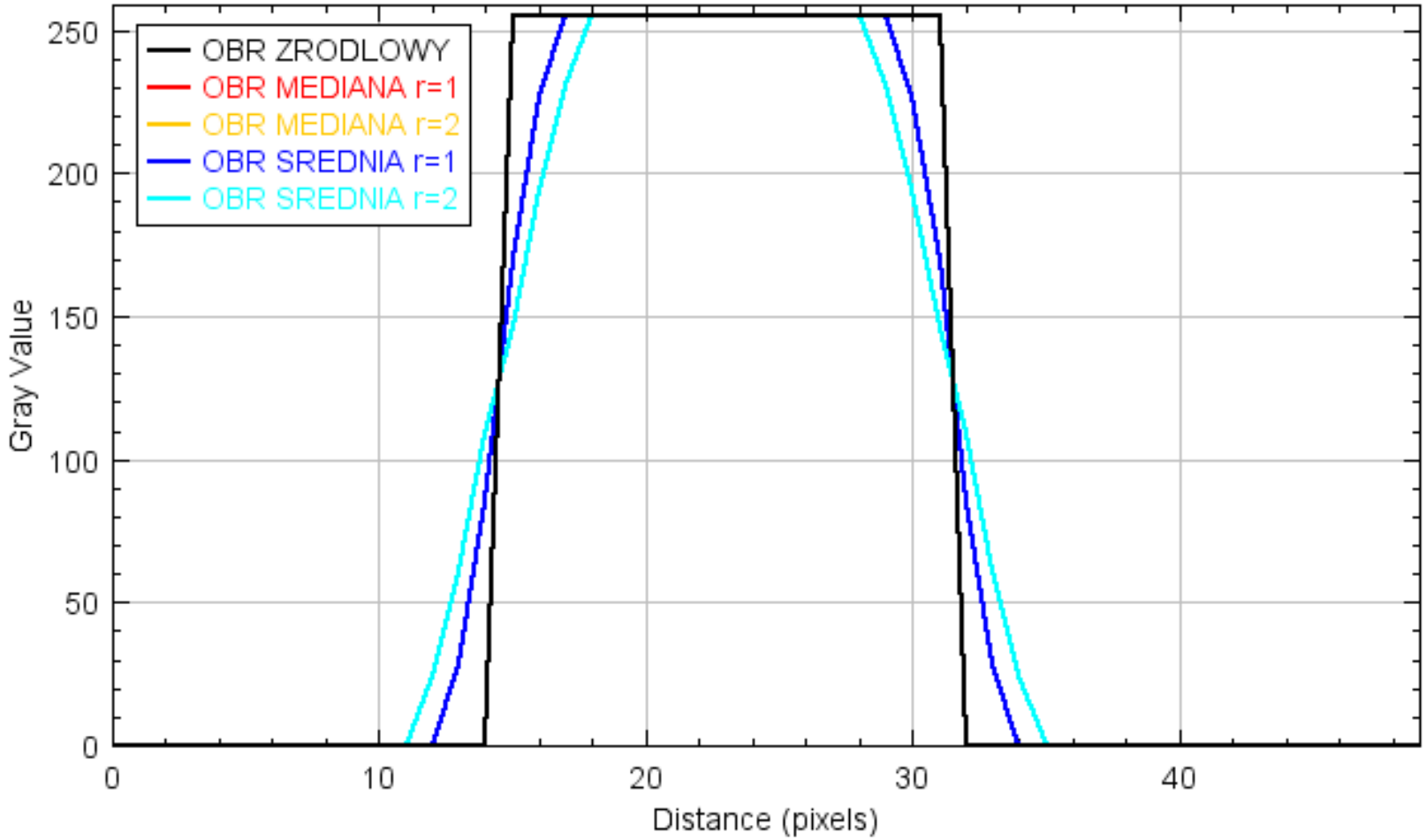
średnia r=2

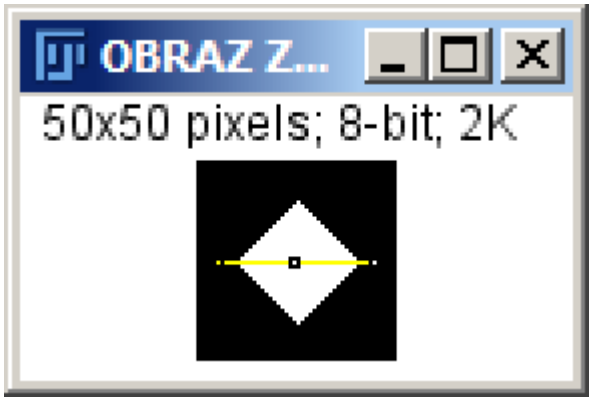


N: 2500
Mean: 49.060
StdDev: 92.203
Value: 150

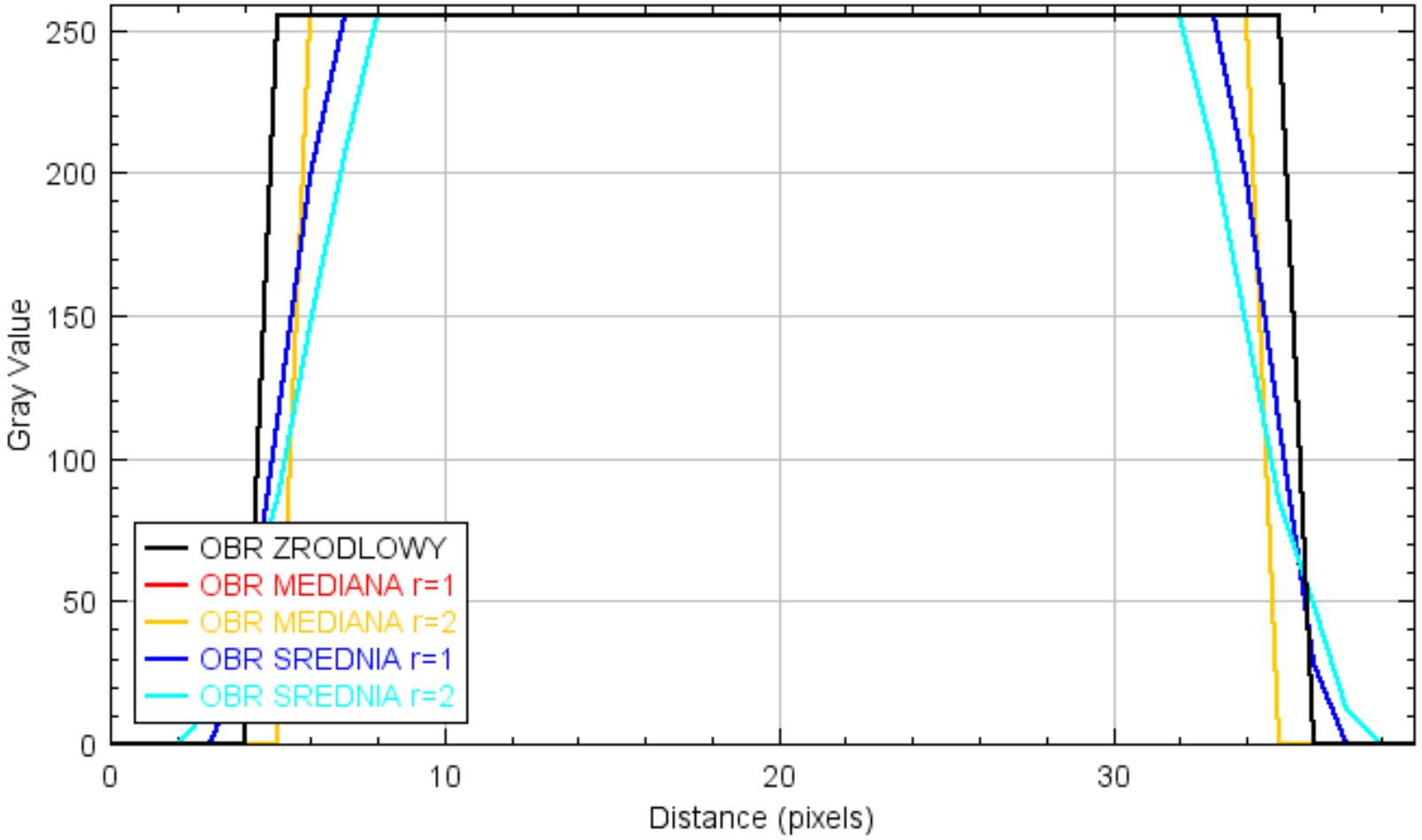
Min: 0
Max: 255
Mode: 0 (1819)
Count: 0

Mediana a uśrednianie

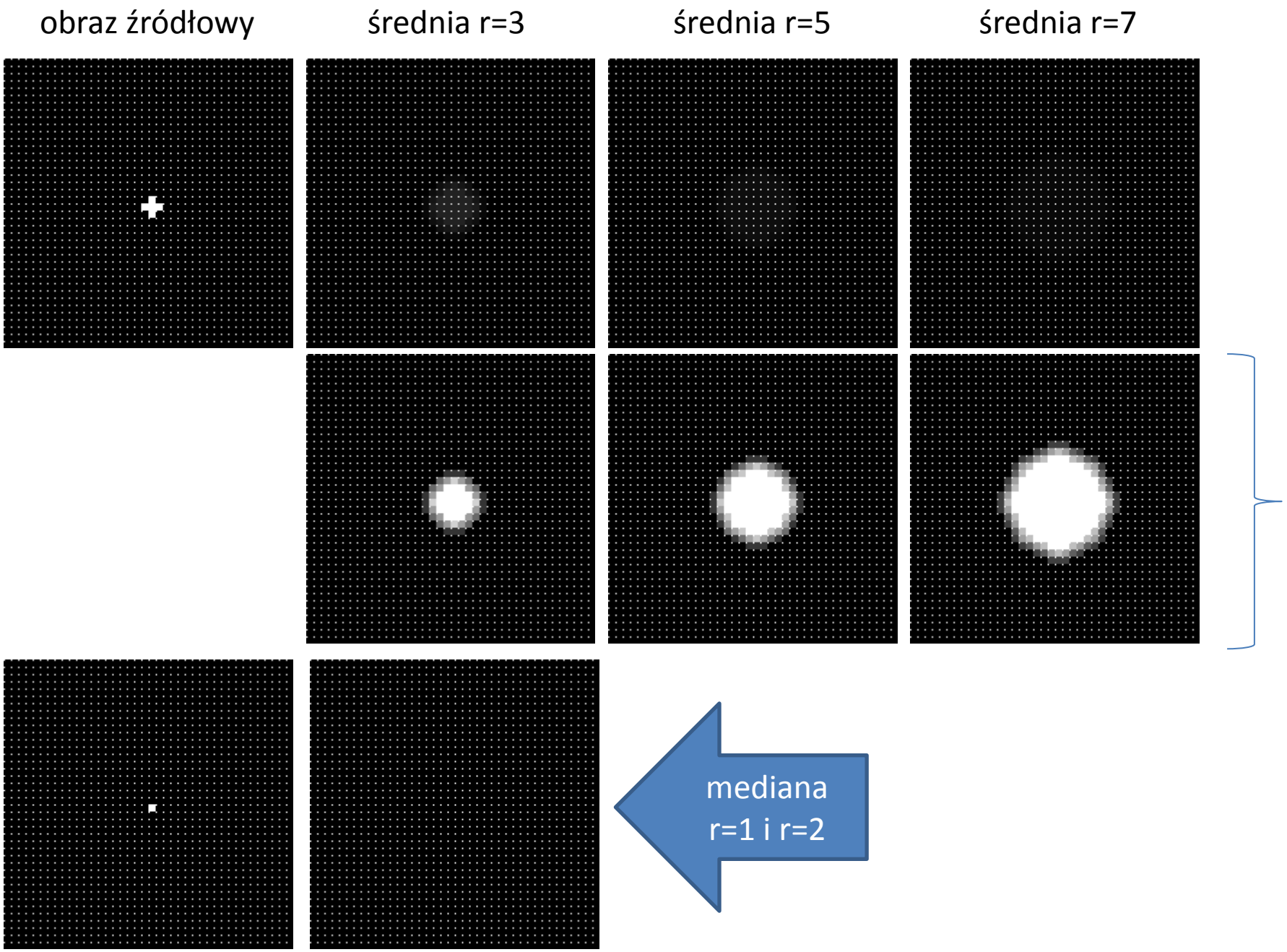




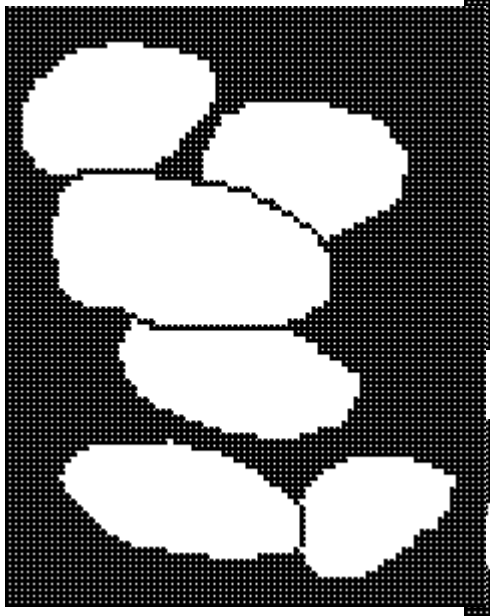
Mediana a uśrednianie

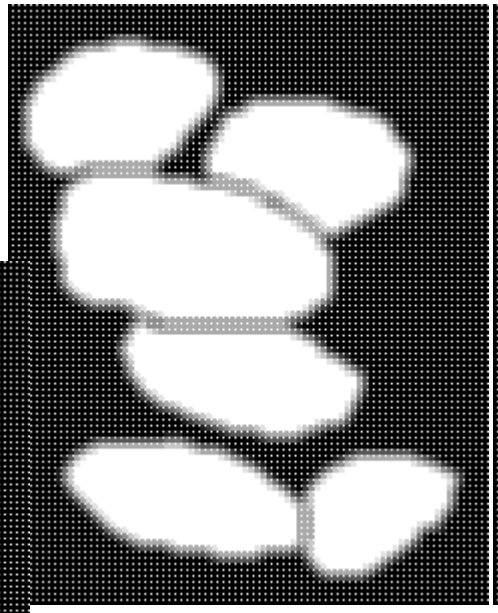


Mediana a uśrednianie - sąsiedztwo

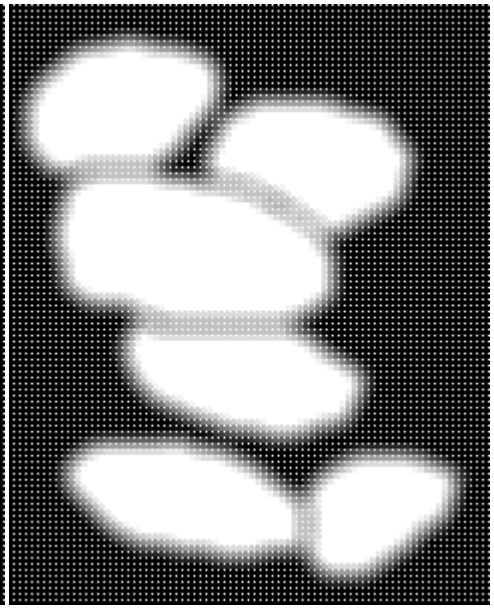


Mediana a uśrednianie - sąsiedztwo

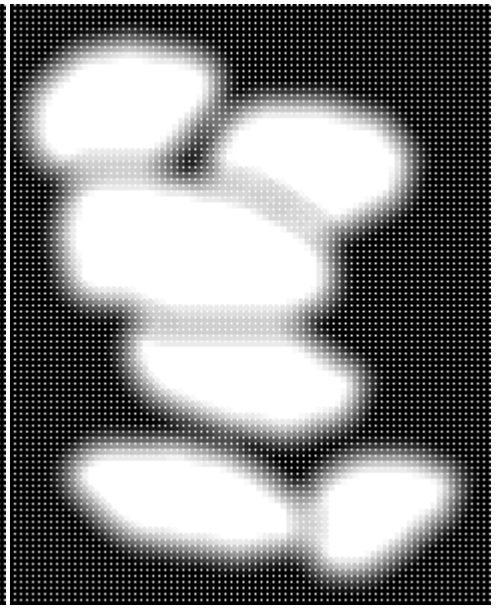




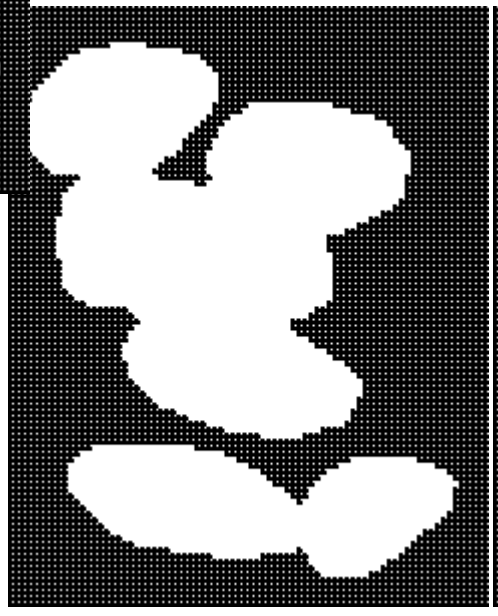
r=1

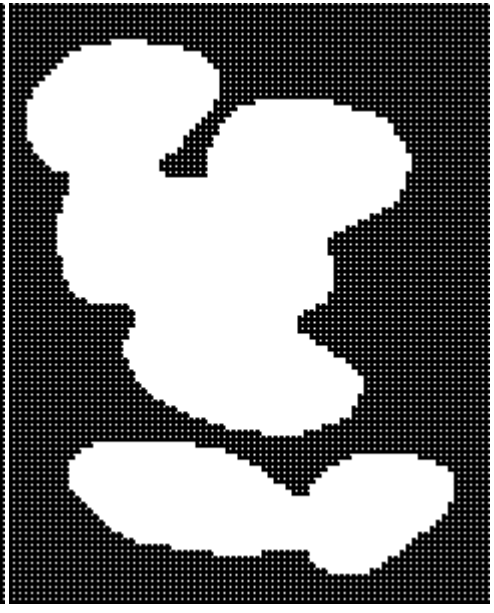


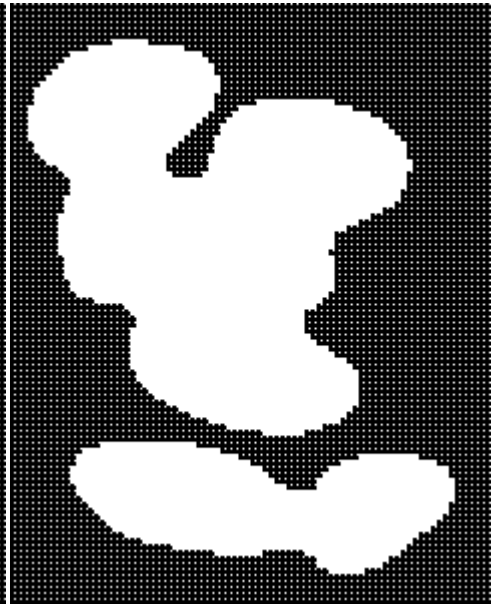
r=2



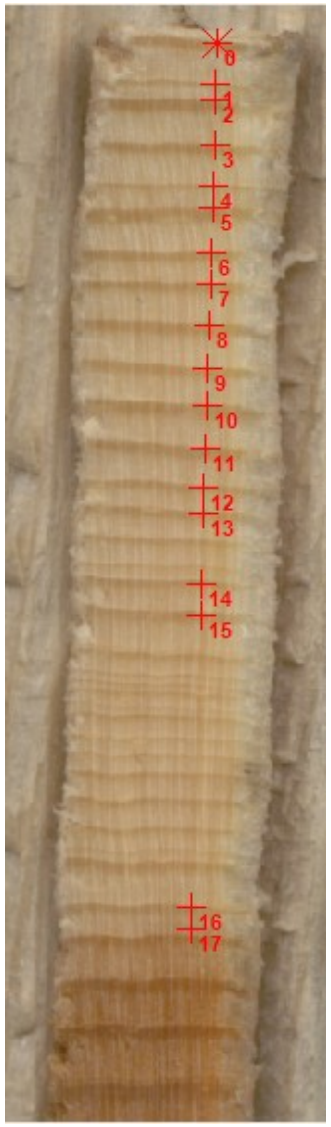
r=3





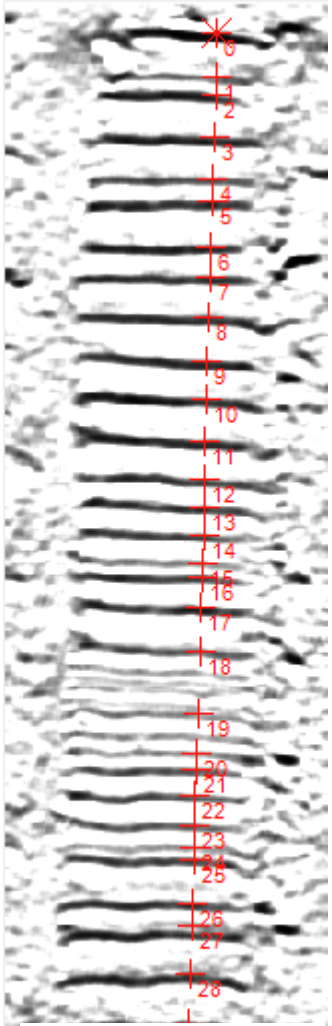


Filtracja kierunkowa

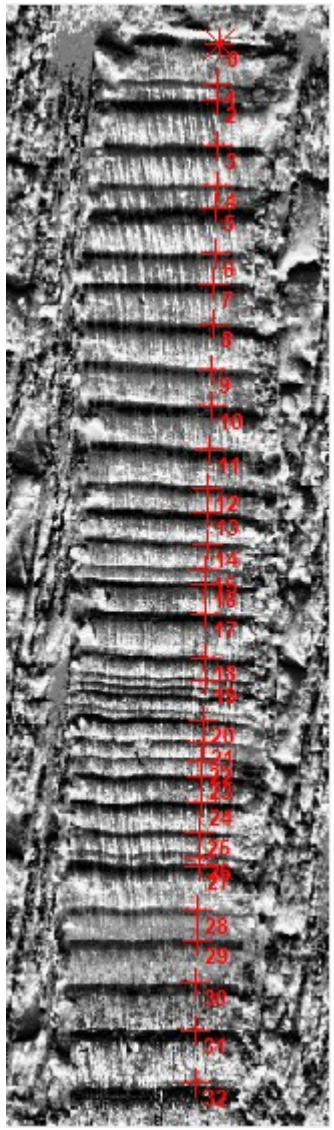


program X

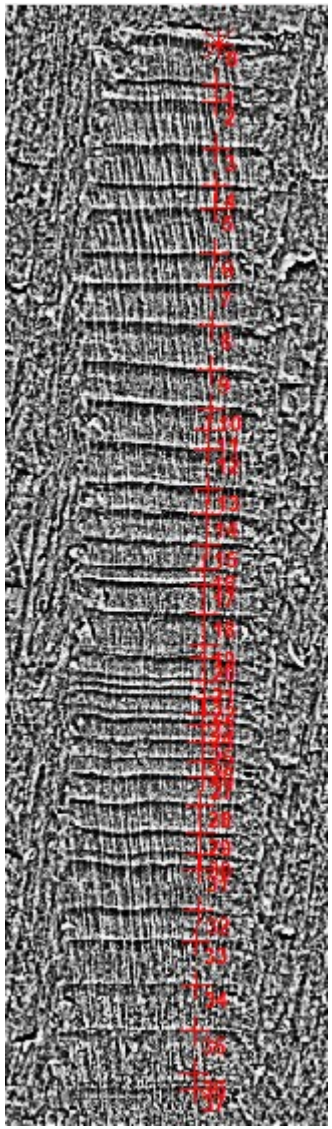
ręcznie



$$\begin{bmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -2 & -2 & -2 & -2 & -2 & -2 & -2 \\ -3 & -3 & -3 & -3 & -3 & -3 & -3 \end{bmatrix}$$



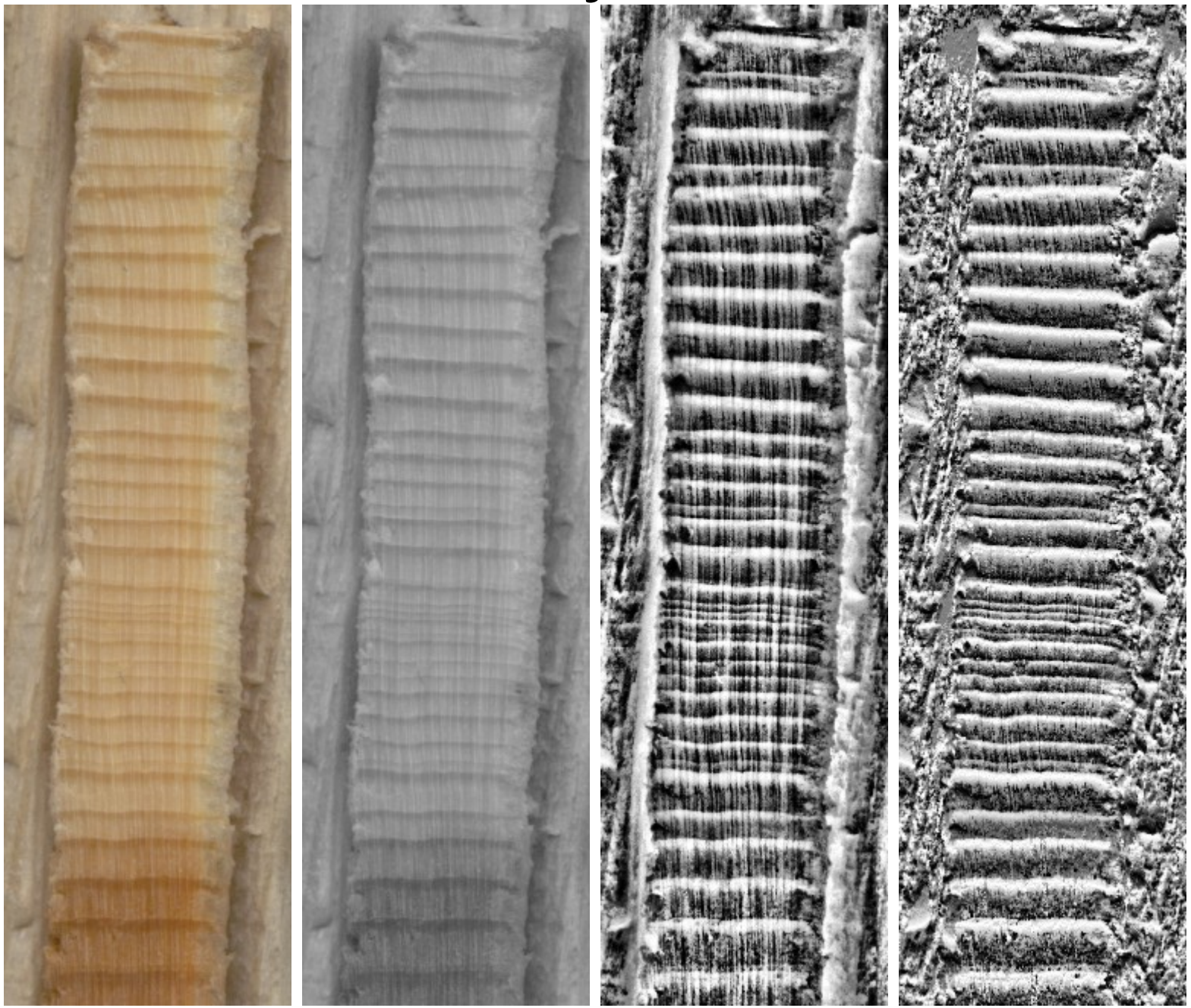
SDA, r=30H



norm r=30
SDA, r=30H

Fabijańska A., Danek M., Barniak J., Piórkowski A.: A Comparative Study of Image -Ring Analysis. IP&C 2016. Springer, 2017, AISC, vol 525, pp 69-78

Filtracja kierunkowa



(3)
dla maski
koło,
r=20

(4)
dla maski
pionowej,
r=20

Filtry oszczędzające krawędzie

- Sigma Filter
- Non-Local Means
- Kuwahara
- Bilateral Filter
- Anisotropic Diffusion
- …

Lokalne wyrównywanie histogramu

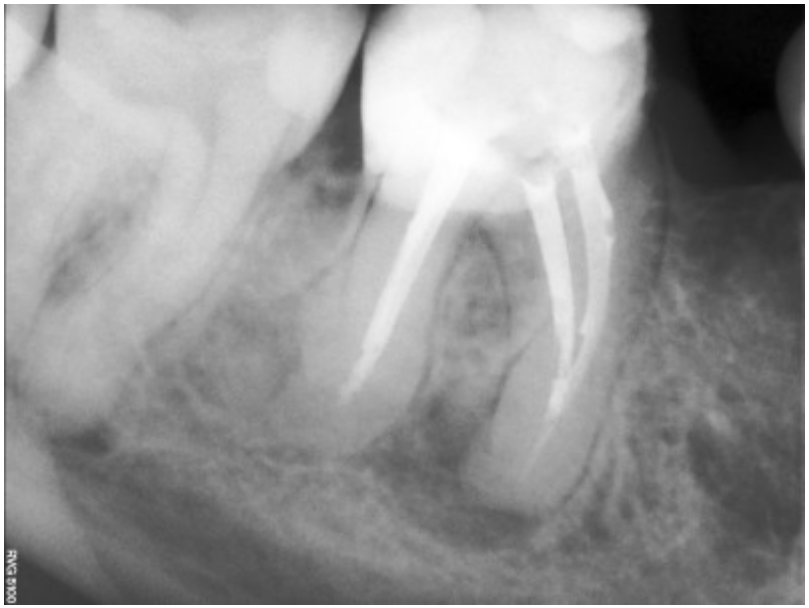
- wykonywane w zadanym sąsiedztwie (punkt centralny histogramu),
- AHE – Adaptive Histogram Equalization
- CLAHE – Contrast Limited AHE

- jest parametryczne (zależy od rozmiaru regionu)
- zwiększa kontrast krawędzi
- zwiększa udział szumu …

Autorska kombinacja ☺

src

HIST EQ



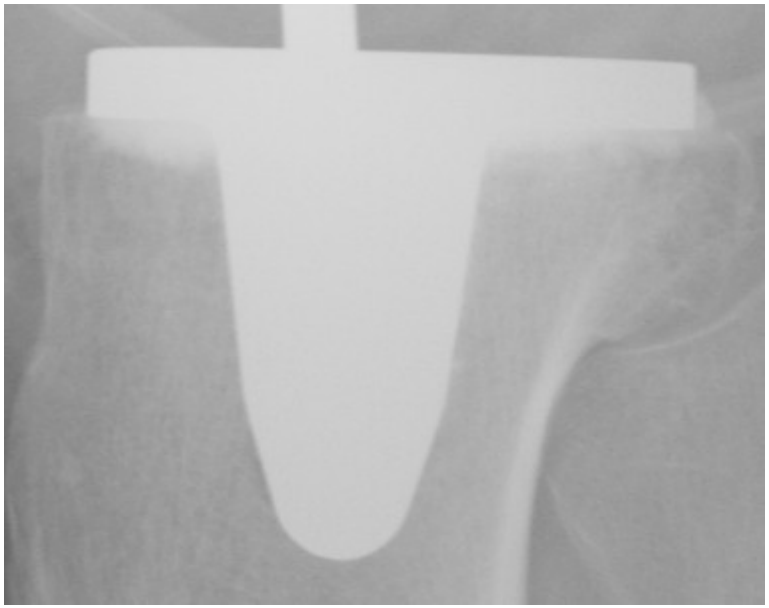
CLAHE

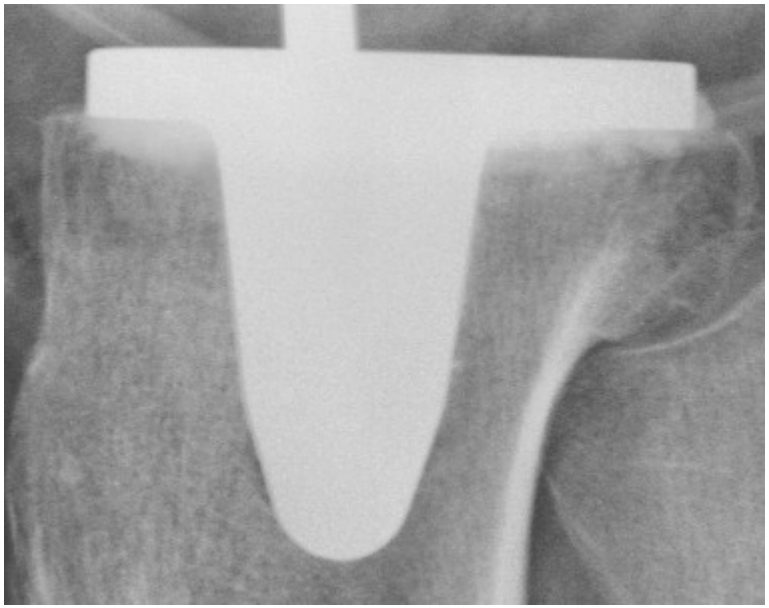
CLAHE -> HISTEQ

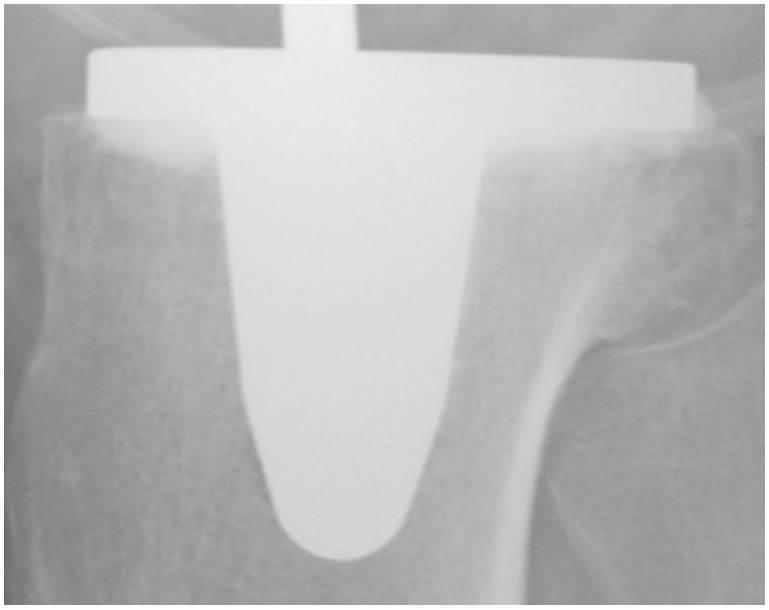
Łebska J., Obuchowicz B., Obuchowicz R., Piórkowski A.: Comparison of the Effects of Different Preprocessing Methods on Homogeneity Assessment of Digital Intraoral Radiographs of Root Canal Fillings. IP&C 2021. Springer, 2022, LNNS, vol 255, pp 145-152

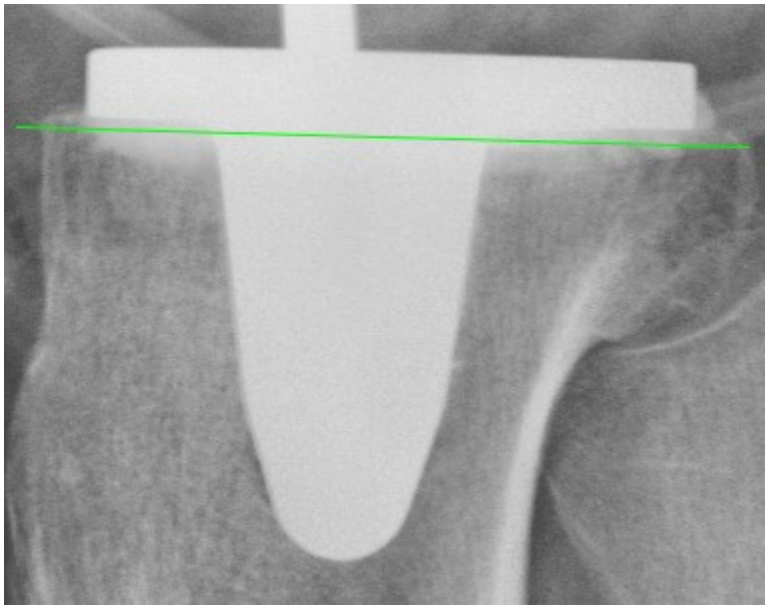
SRC

CLAHE



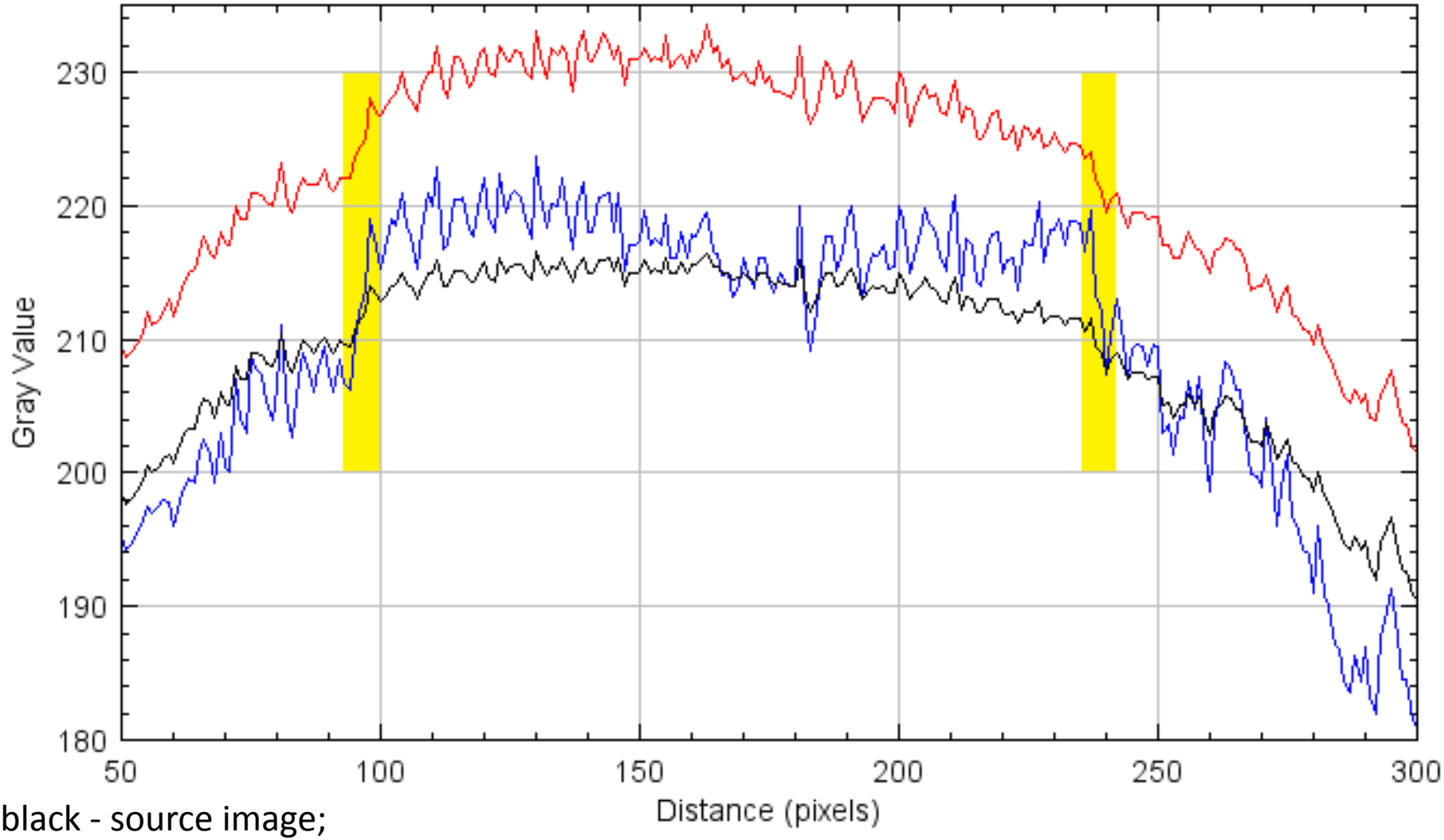






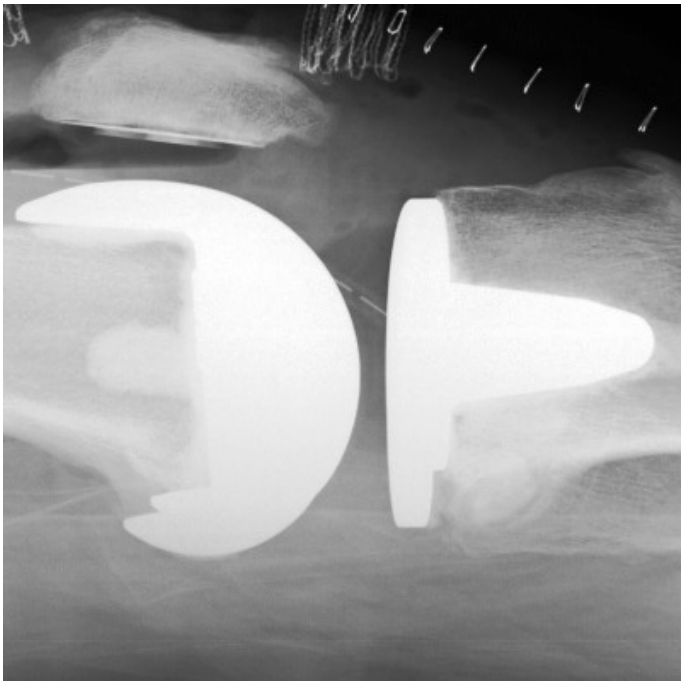
HIST EQ

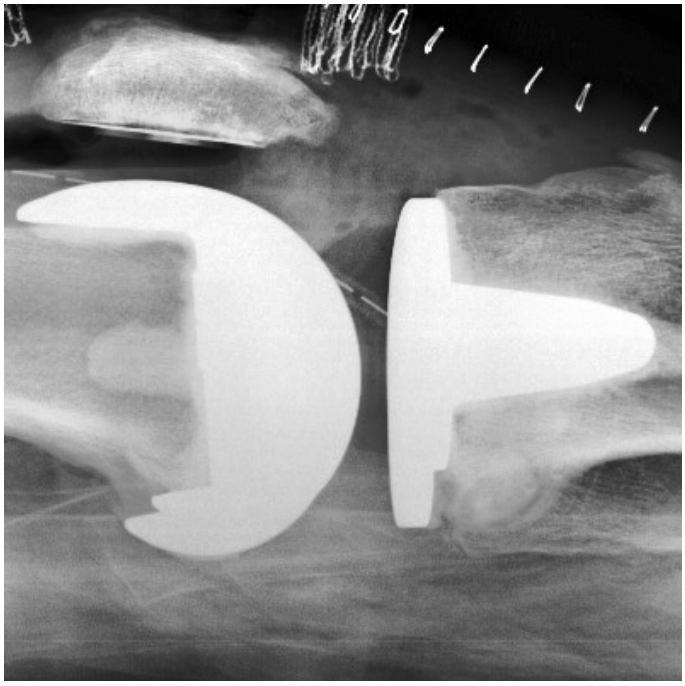
Uwaga! Powyższe obrazy są jedynie fragmentami pełnego radiogramu



black - source image;
red - after histogram equalization;
blue - after CLAHE;
yellow zone - the signal slope on the edges of the implant.

Smolikowska A., Kamiński P., Obuchowicz R., Piórkowski A.: Effect of Adaptive Histogram Equalization of Orthopedic Radiograph Images on the Accuracy of Implant Delineation Using Active Contours. Springer, 2021, LNNS vol 285, pp 136-144.



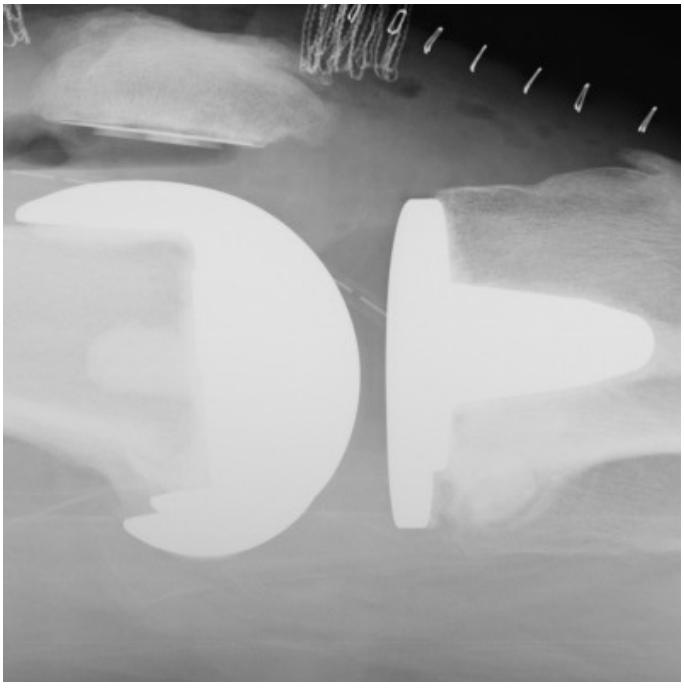


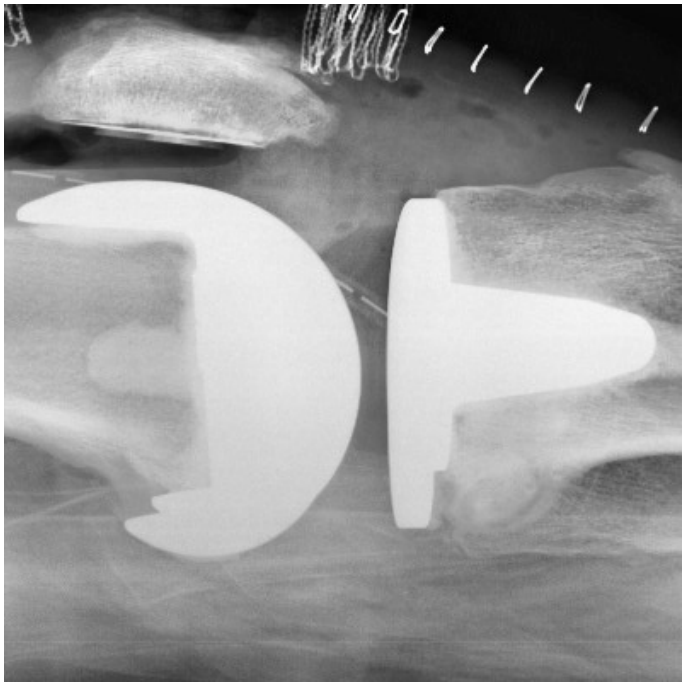
HIST EQ

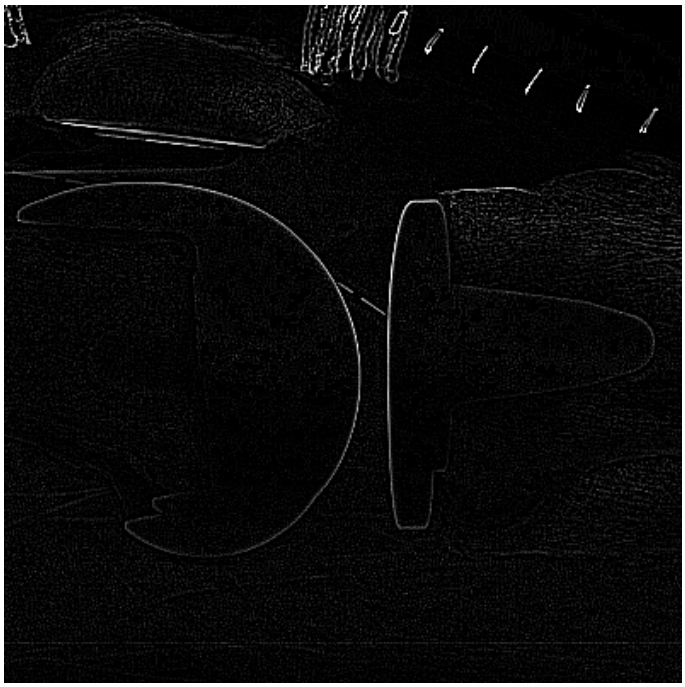
CLAHE -> HISTEQ

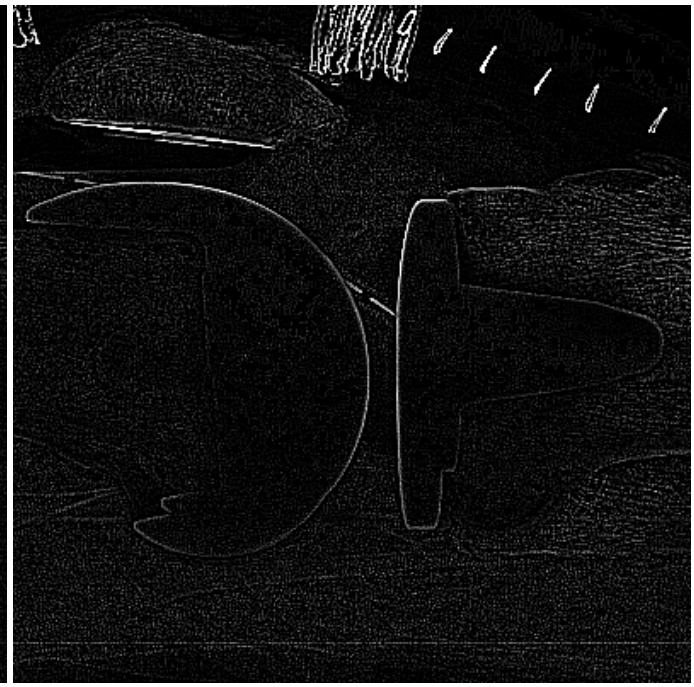
src

CLAHE









-1	-1	-1
-1	8	-1
-1	-1	-1

HIST EQ

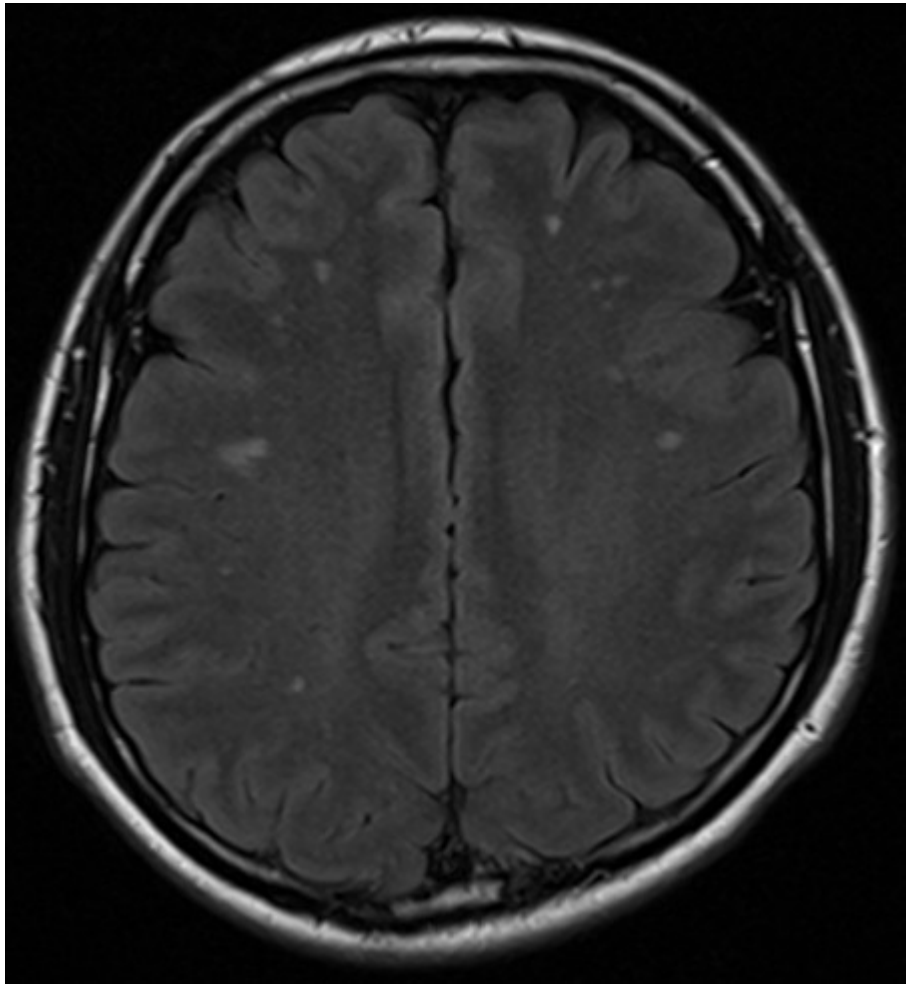
src

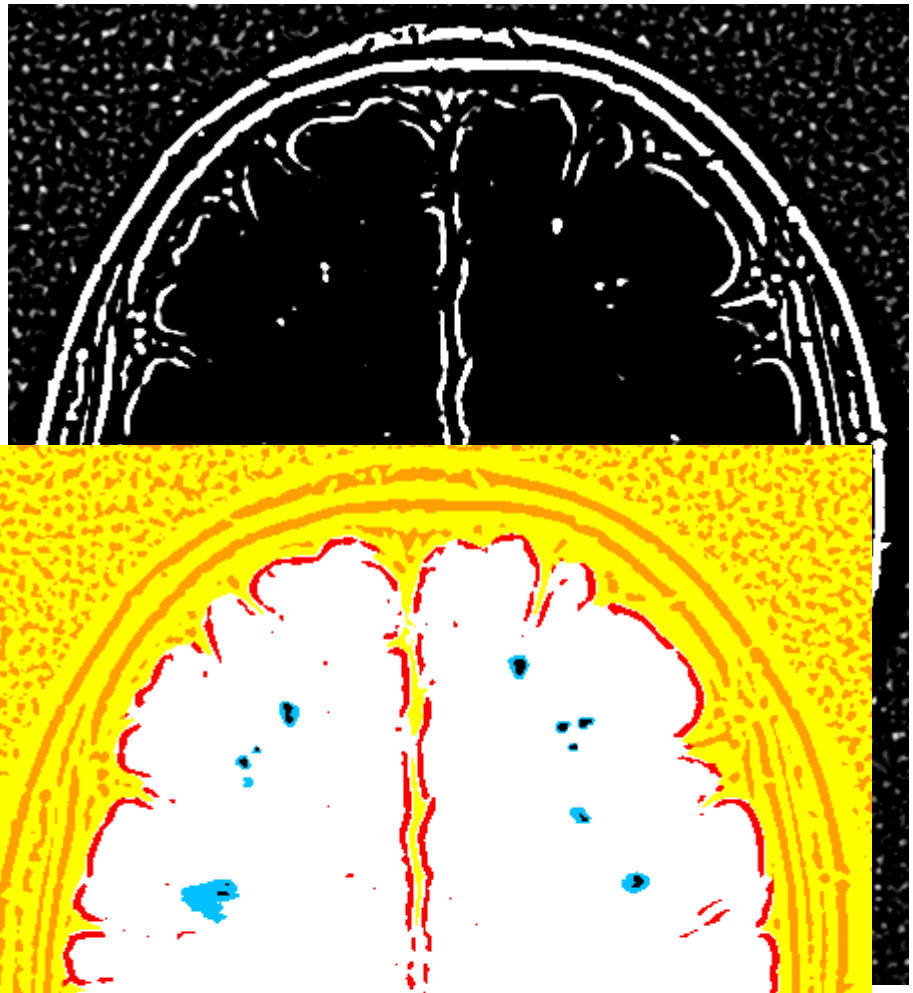
CLAHE -> HISTEQ

CLAHE

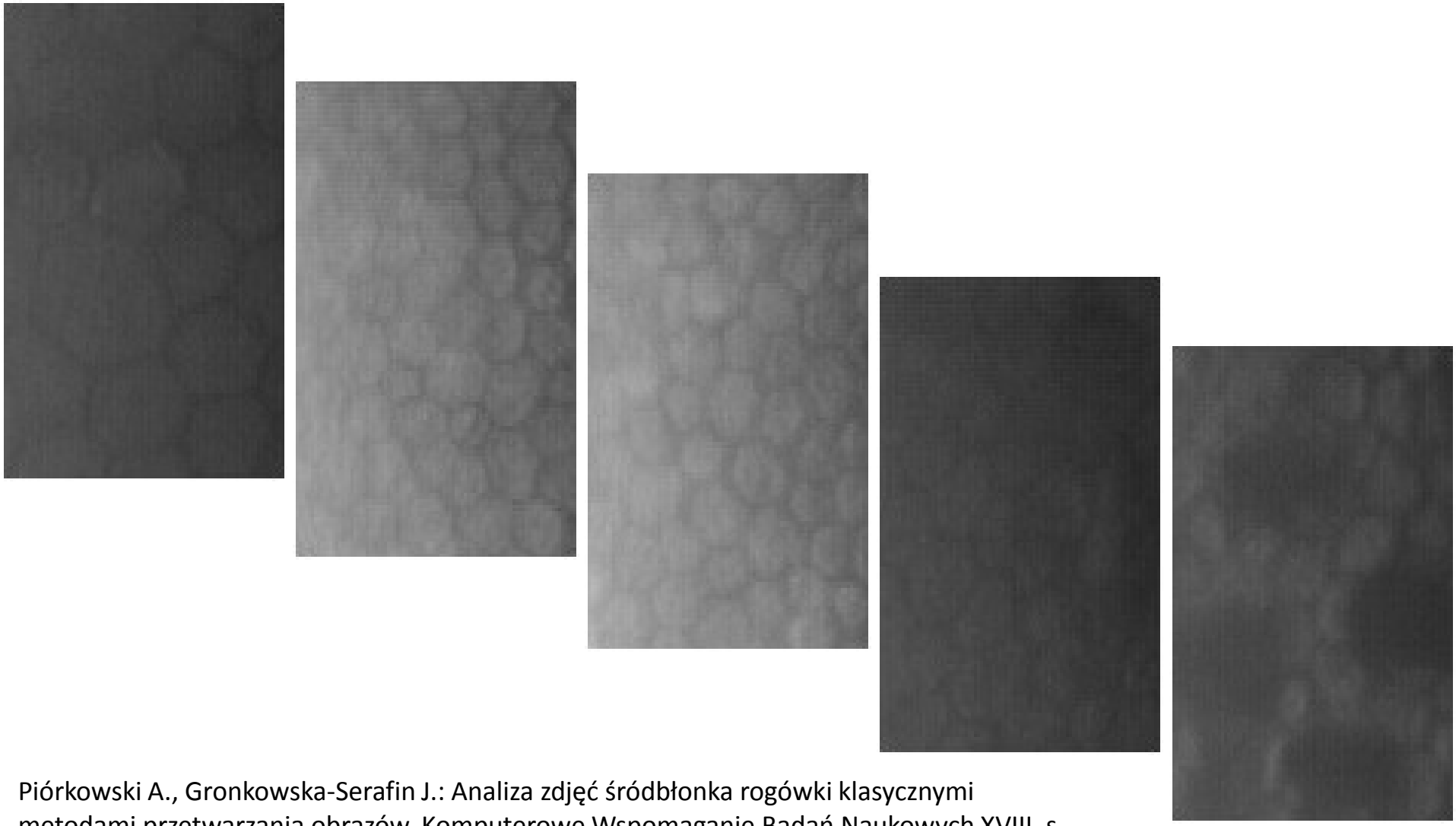
Dedykowane zastosowania filtrów konwolucyjnych

-3	-3	-2	-2	-2	-2	-2	-3	-3
-3	-2	-1	-1	-1	-1	-1	-2	-3
-2	-1	0	0	0	0	0	-1	-2
-2	-1	0	6	6	6	0	-1	-2
-2	-1	0	6	40	6	0	-1	-2
-2	-1	0	6	6	6	0	-1	-2
-2	-1	0	0	0	0	0	-1	-2
-3	-2	-1	-1	-1	-1	-1	-2	-3
-3	-3	-2	-2	-2	-2	-2	-3	-3





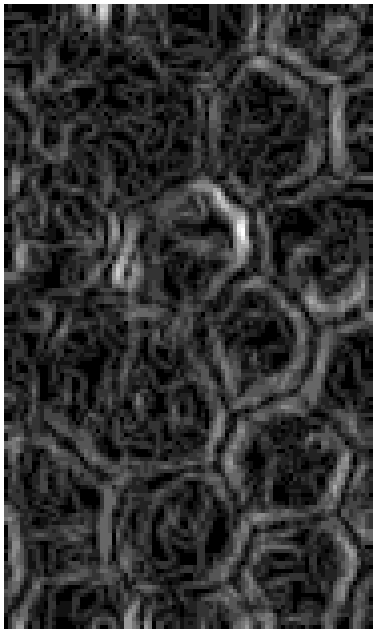
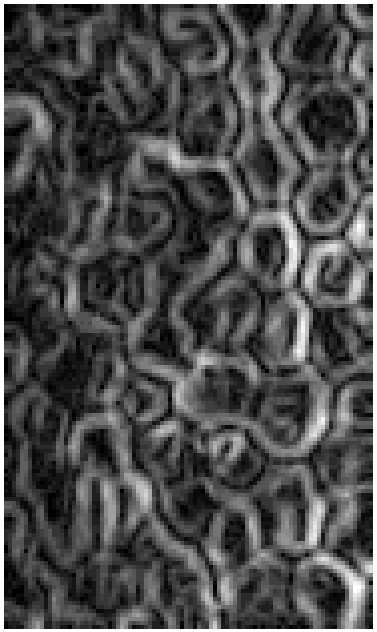
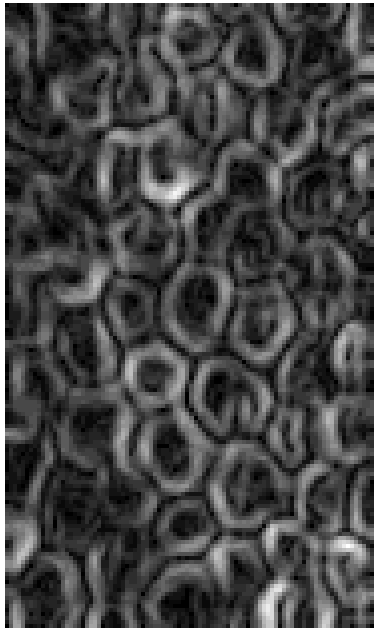
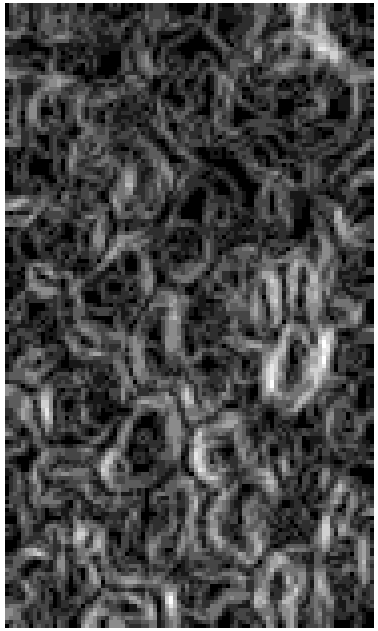
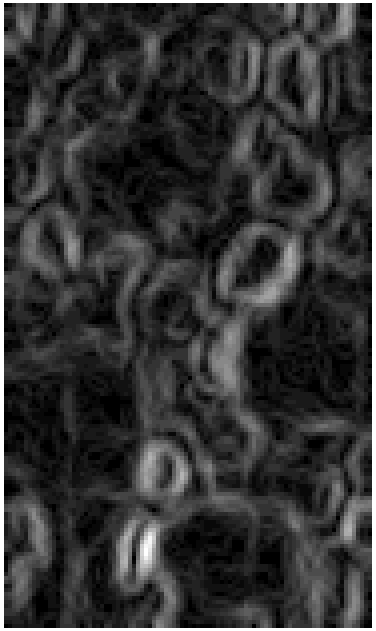
Obrazy śródbłonka rogówki



Piórkowski A., Gronkowska-Serafin J.: Analiza zdjęć śródbłonka rogówki klasycznymi metodami przetwarzania obrazów. Komputerowe Wspomaganie Badań Naukowych XVIII, s. 283-290, Wrocławskie Towarzystwo Naukowe, 2011.

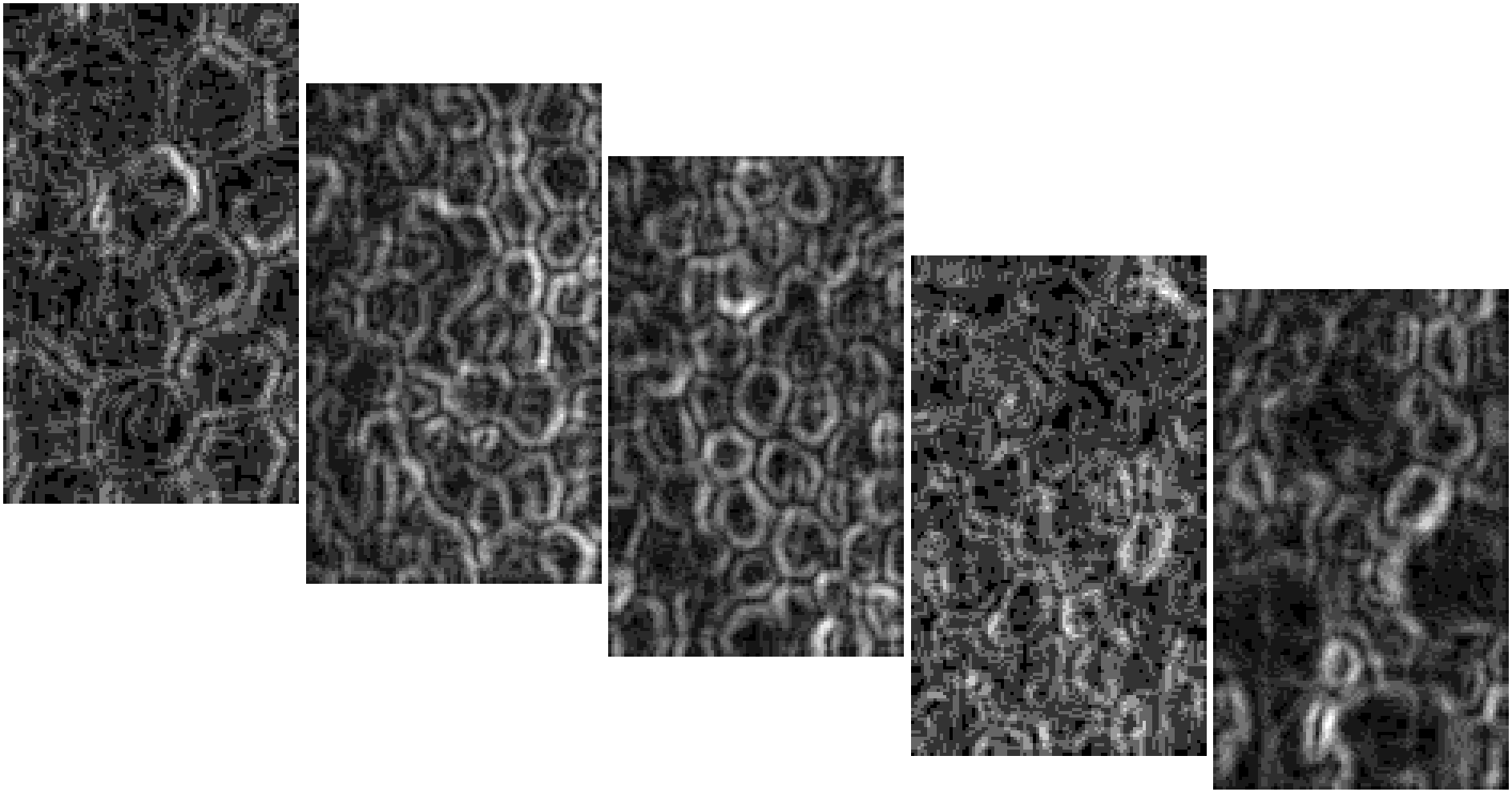
Obrazy śródbłonka rogówki

ZŁOŻENIE FILTRÓW PREWITTA

Obrazy śródbłonka rogówki

GRADIENT MORFOLOGICZNY



Obrazy śródbłonka rogówki

WYODRĘBNIANIE KOMÓREK – ‘CAPRICORN’

1	1	1	-1	-1	-1	1	1	1
1	1	1	-1	-1	-1	1	1	1
1	1	1	-1	-1	-1	1	1	1
1	1	1	-2	-2	-2	1	1	1
1	1	-1	-2	-2	-2	-1	1	1
1	-1	-1	-2	-2	-2	-1	-1	1
-1	-1	-1	1	2	1	-1	-1	-1
-1	-1	1	1	1	1	1	-1	-1
-1	1	1	1	0	1	1	1	-1

-1	1	1	1	0	1	1	1	-1
-1	-1	1	1	1	1	1	-1	-1
-1	-1	-1	1	2	1	-1	-1	-1
1	-1	-1	-2	-2	-2	-1	-1	1
1	1	-1	-2	-2	-2	-1	1	1
1	1	1	-2	-2	-2	1	1	1
1	1	1	-1	-1	-1	1	1	1
1	1	1	-1	-1	-1	1	1	1
1	1	1	-1	-1	-1	1	1	1

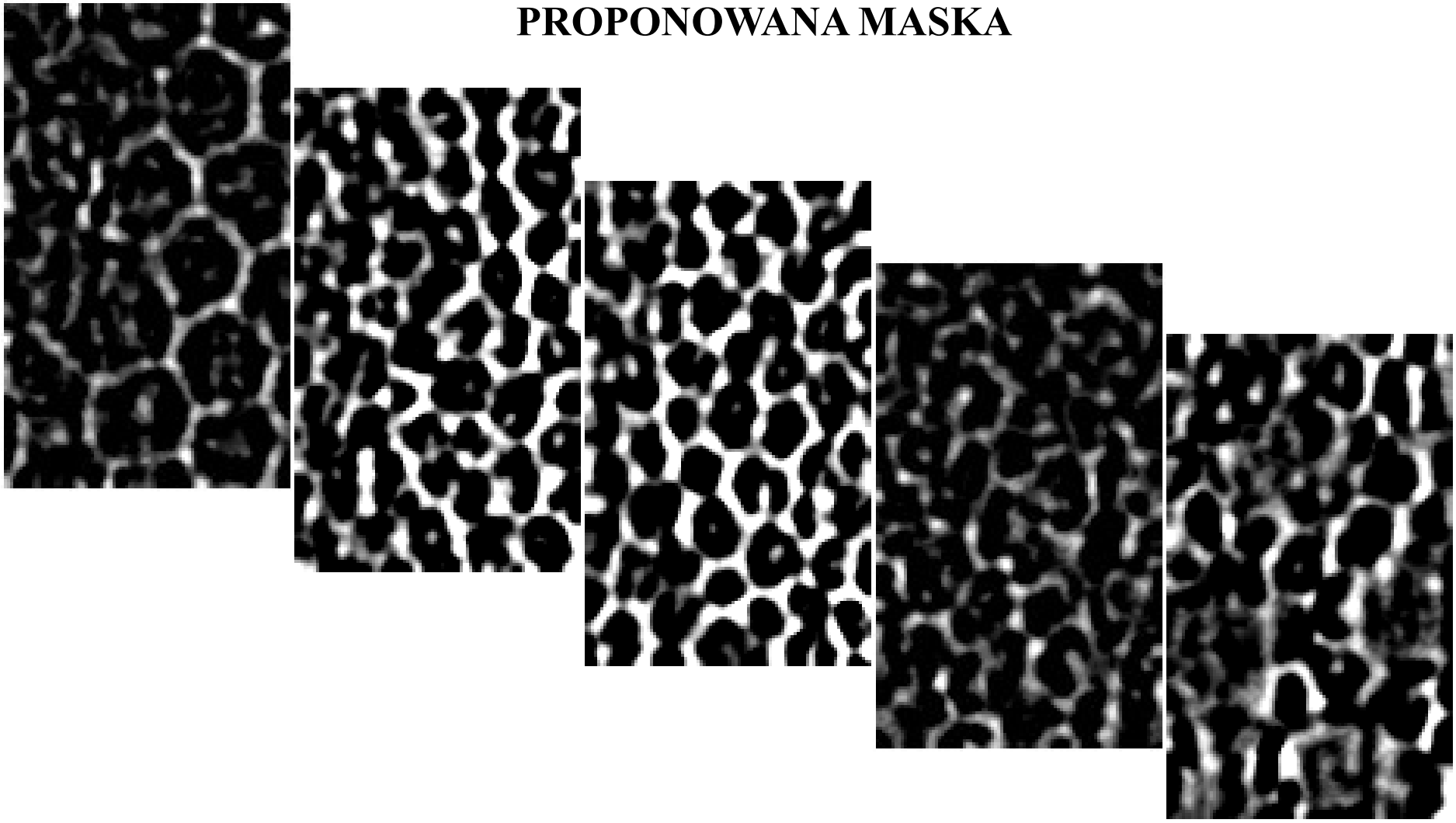
Mahzoun M. R., Okazaki K., Mitsumoto H., Kawai H., Sato Y., Tamura S., Kani K.: Detection and Complement of Hexagonal Borders in Corneal Endothelial Cell Image. Medical Imaging Technology, Vol. 14 No. 1, January 1996

Obrazy śródbłonka rogówki

PROPONOWANA MASKA

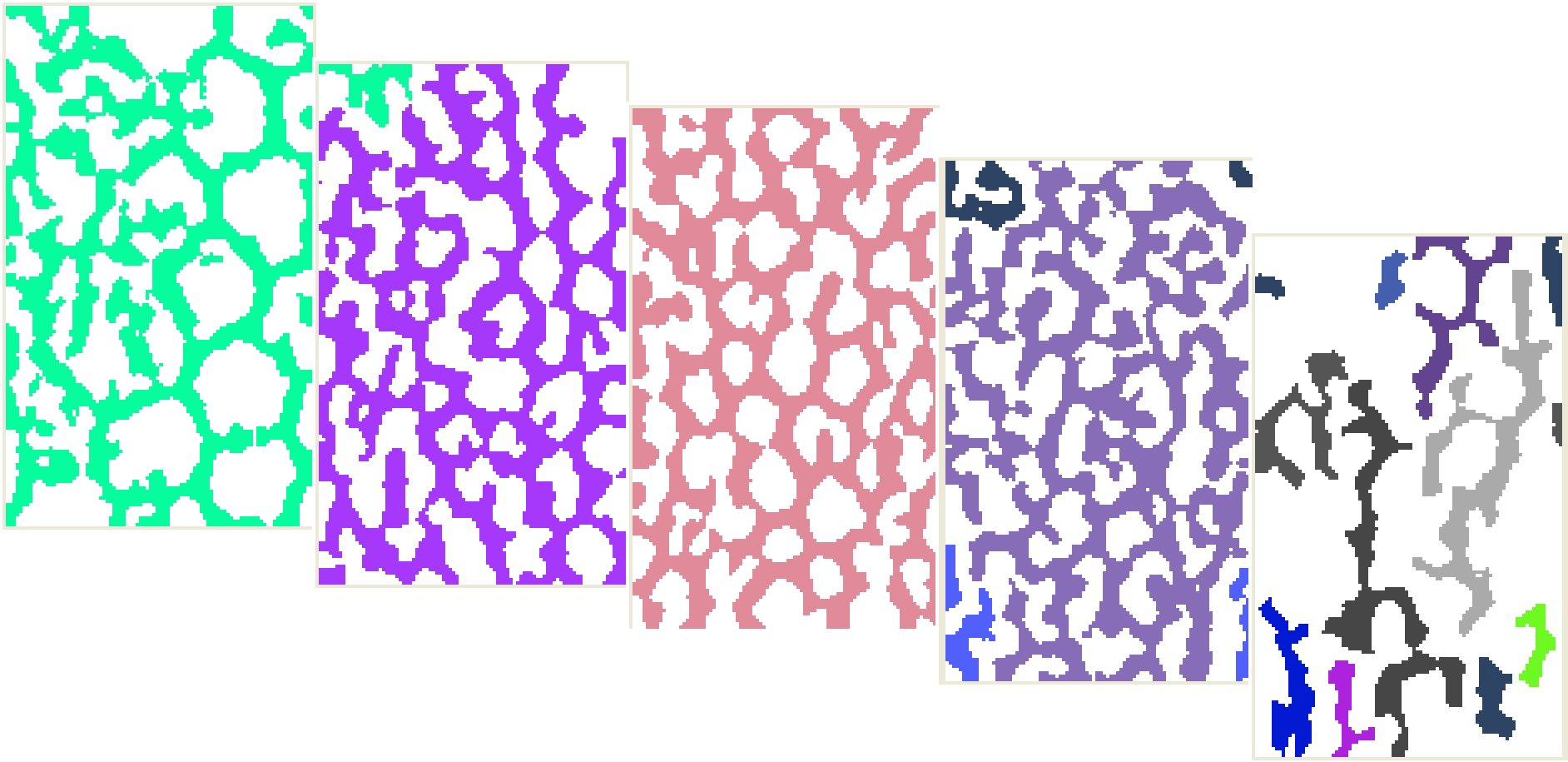
3	3	2	2	2	2	2	3	3
3	2	1	1	1	1	1	2	3
2	1	0	0	0	0	0	1	2
2	1	0	-6	-12	-6	0	1	2
2	1	0	-12	-32	-12	0	1	2
2	1	0	-6	-12	-6	0	1	2
2	1	0	0	0	0	0	1	2
3	2	1	1	1	1	1	2	3
3	3	2	2	2	2	2	3	3

Obrazy śródbłonka rogówki



Obrazy śródbłonka rogówki

INDEKSACJA BRZEGÓW MIĘDZYKOMÓRKOWYCH

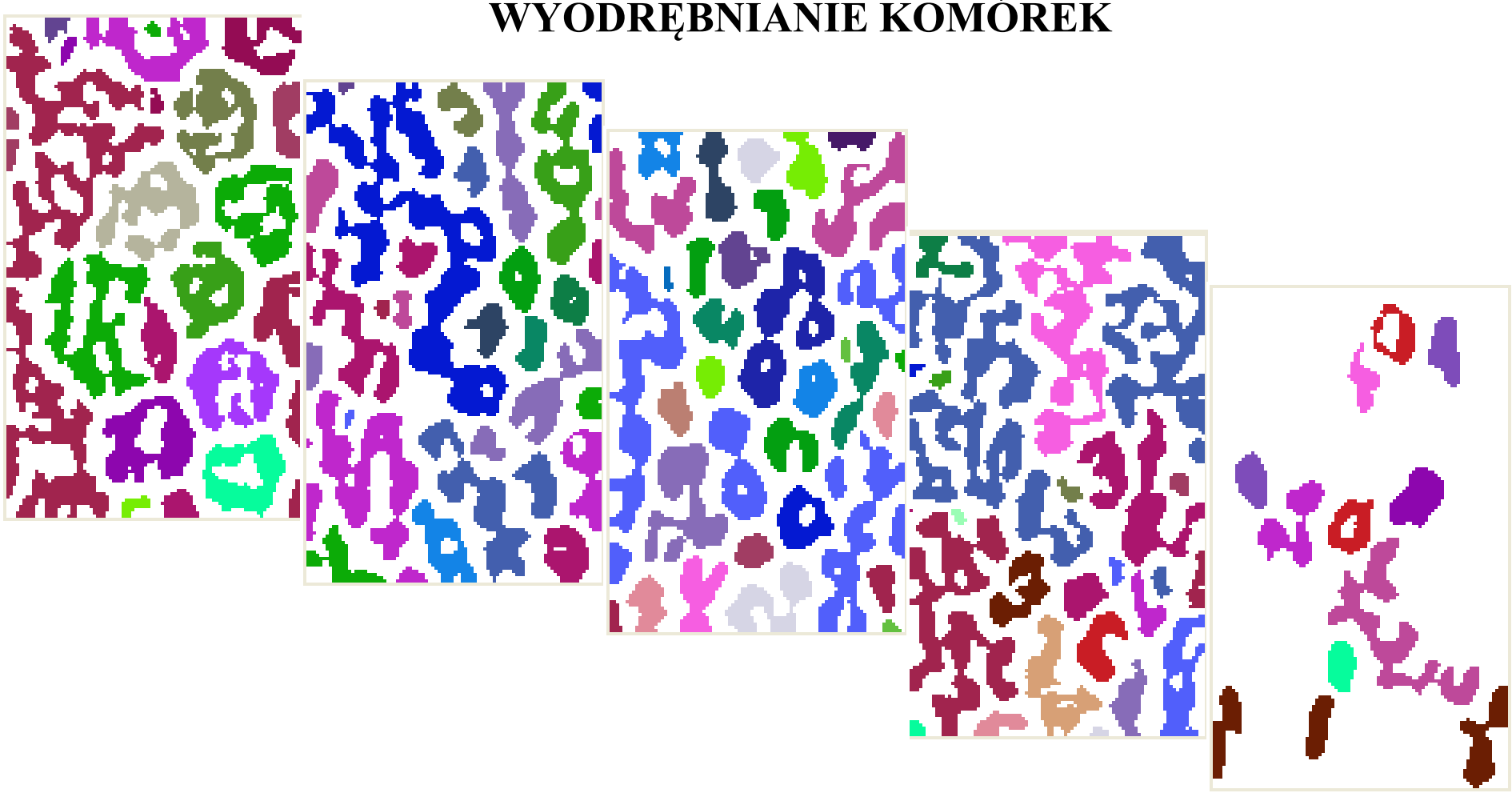


Obrazy śródbłonka rogówki

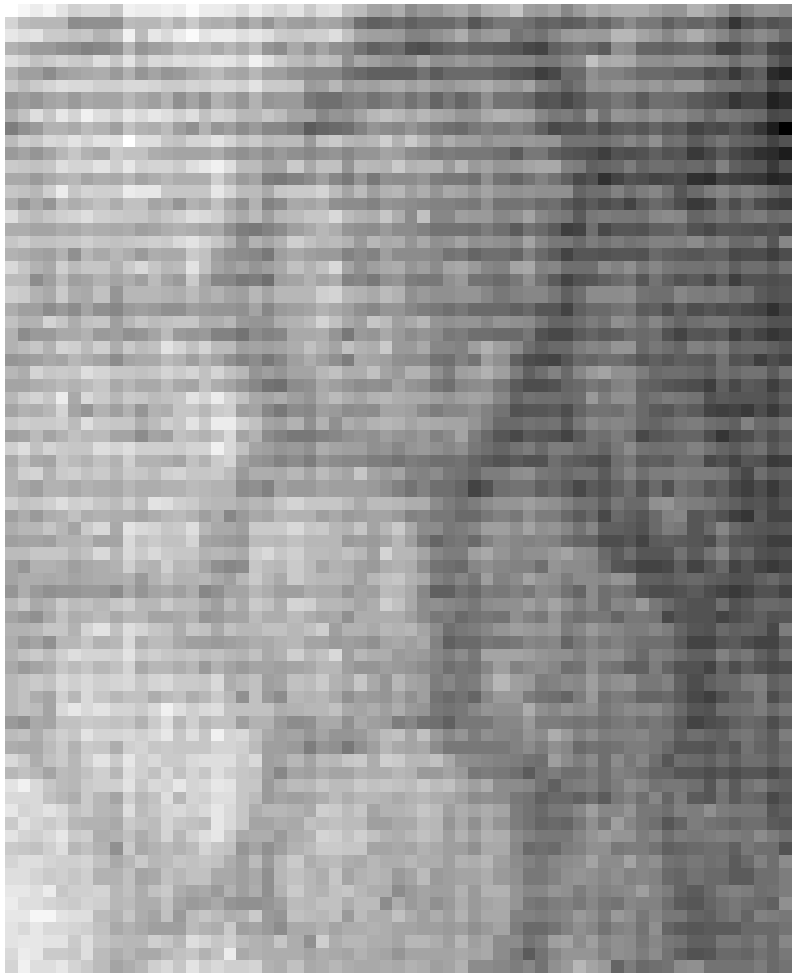
WYODRĘBNIANIE KOMÓREK

-3	-3	-2	-2	-2	-2	-2	-3	-3
-3	-2	-1	-1	-1	-1	-1	-2	-3
-2	-1	0	0	0	0	0	-1	-2
-2	-1	0	6	12	6	0	-1	-2
-2	-1	0	12	32	12	0	-1	-2
-2	-1	0	6	12	6	0	-1	-2
-2	-1	0	0	0	0	0	-1	-2
-3	-2	-1	-1	-1	-1	-1	-2	-3
-3	-3	-2	-2	-2	-2	-2	-3	-3

Obrazy śródbłonka rogówki

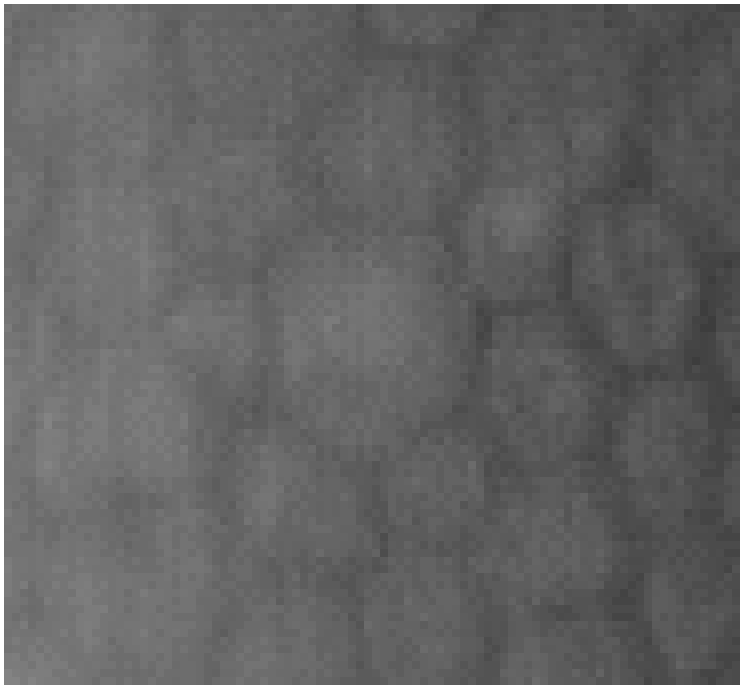


Corneal endothelium images

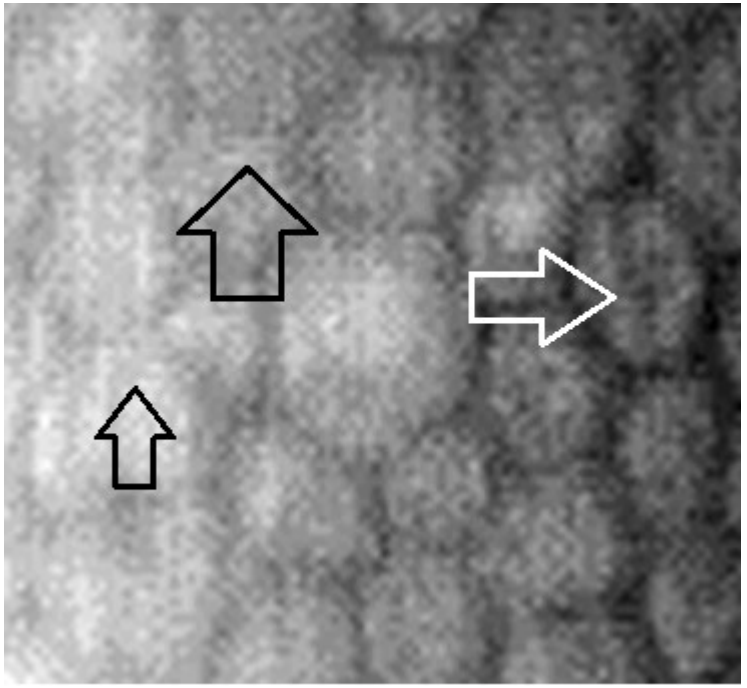


K. Habrat, M. Habrat, J. Gronkowska-Serafin, A. Piorkowski: Cell Detection in Corneal Endothelial Images Using Directional Filters. Springer 2016, AISC Vol. 389, pp 113–123.

Corneal endothelium images

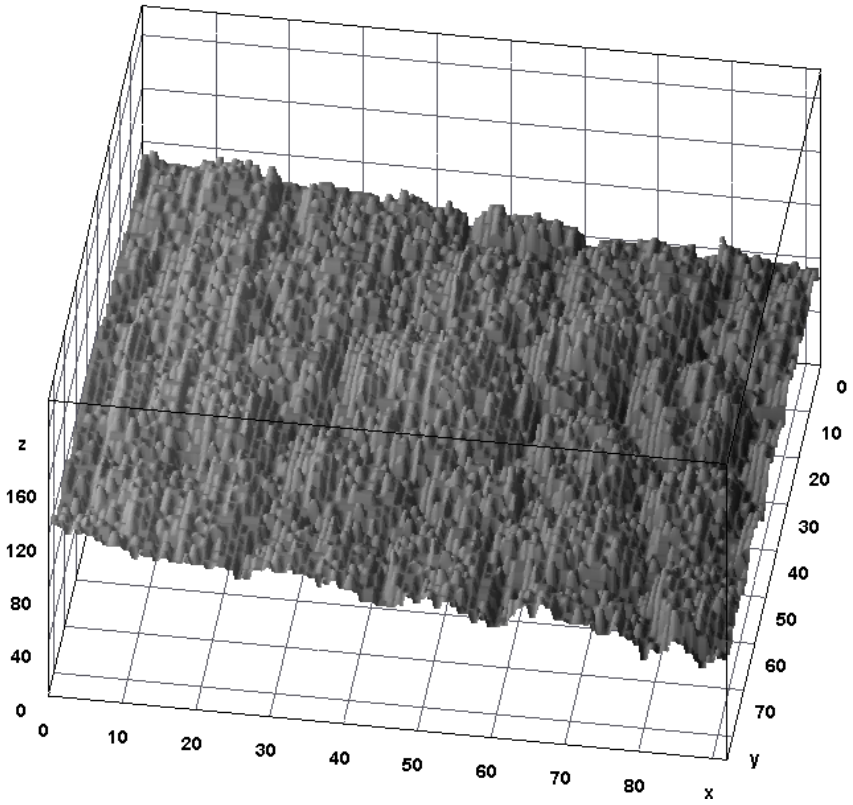


original

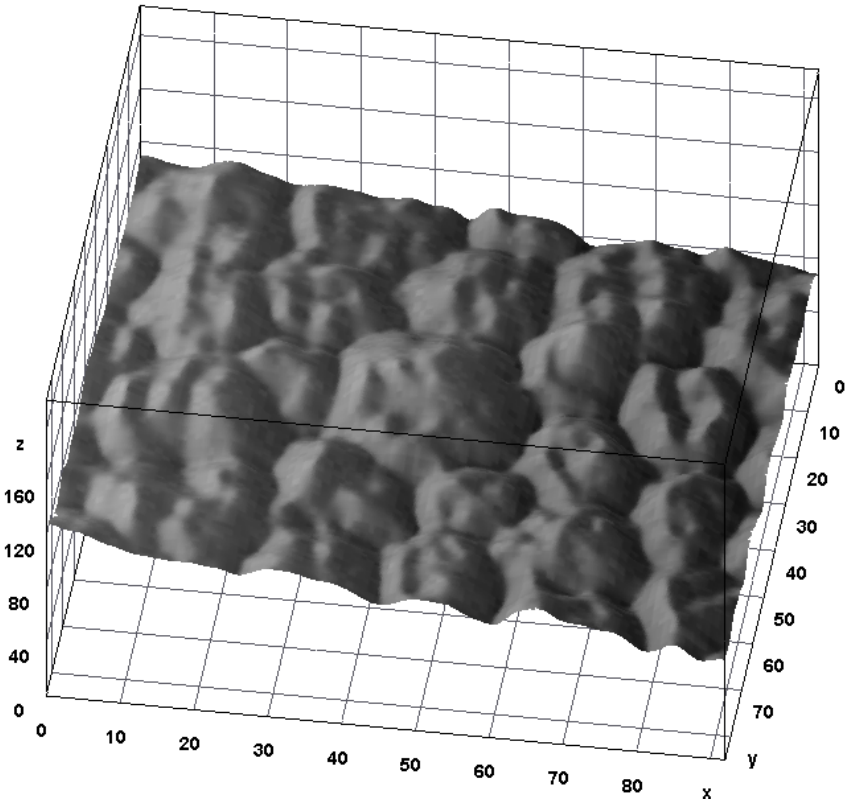


normalized

Corneal endothelium images

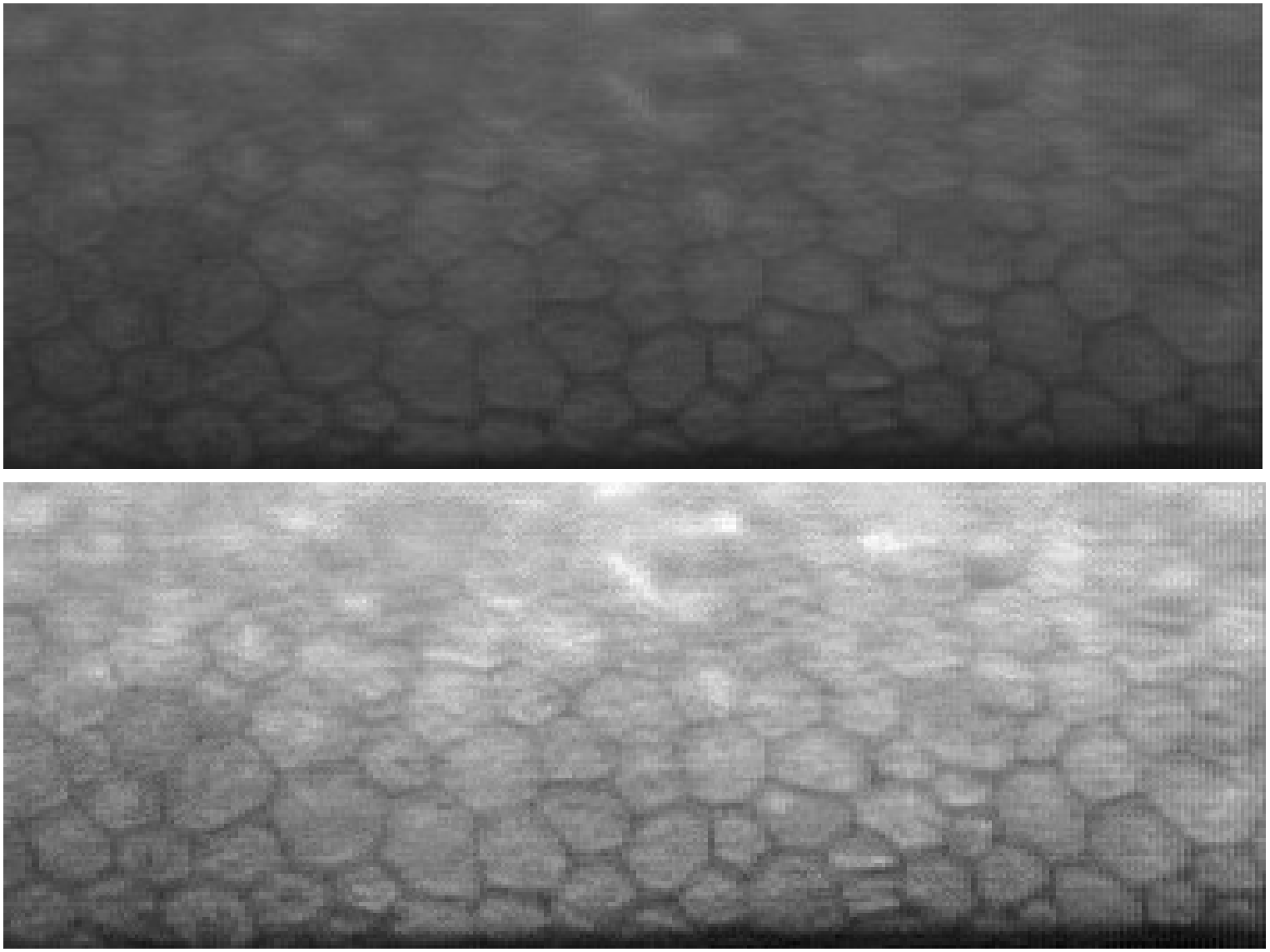


original

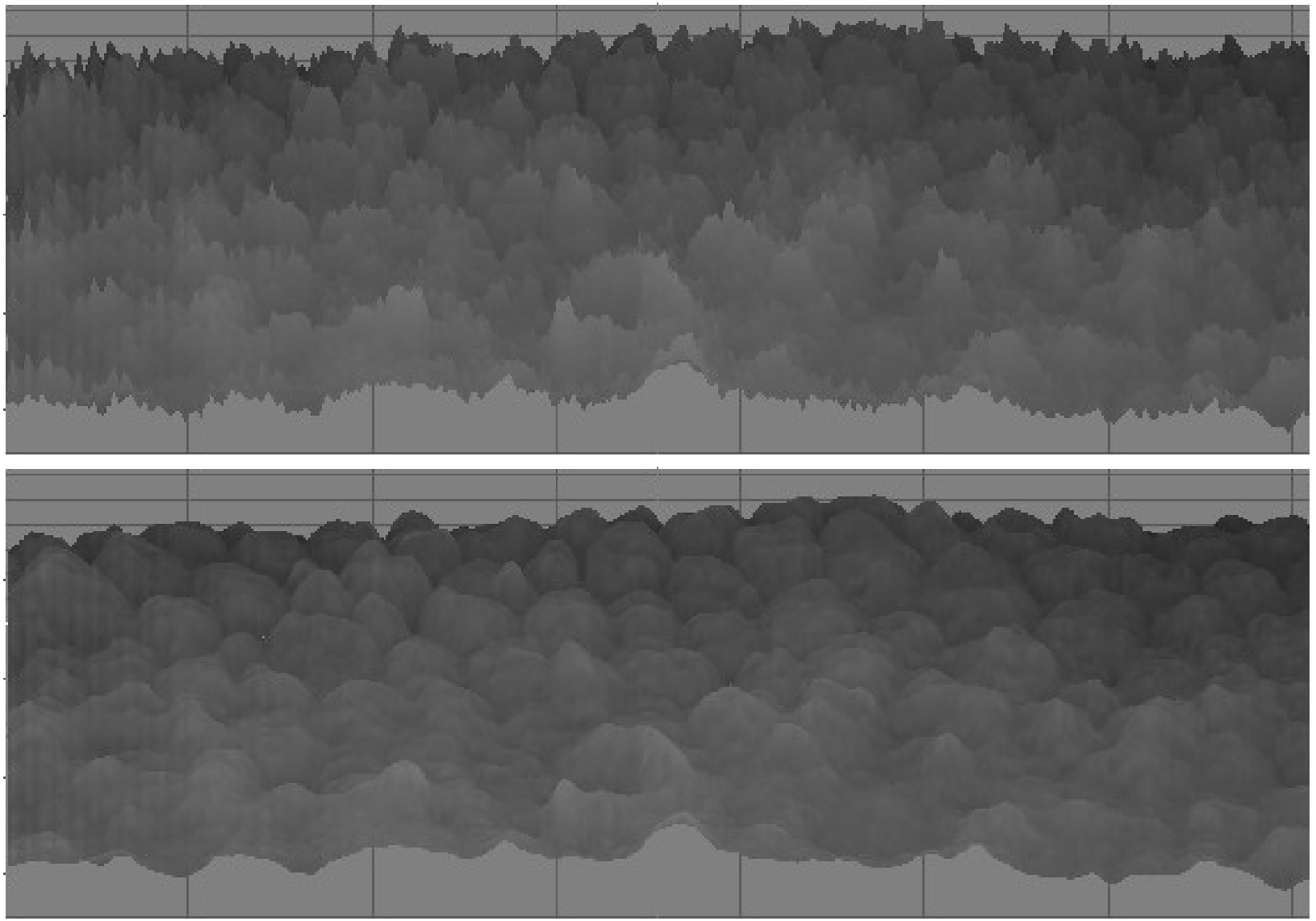


after preprocessing (smoothing)

Corneal endothelium images



Corneal endothelium images



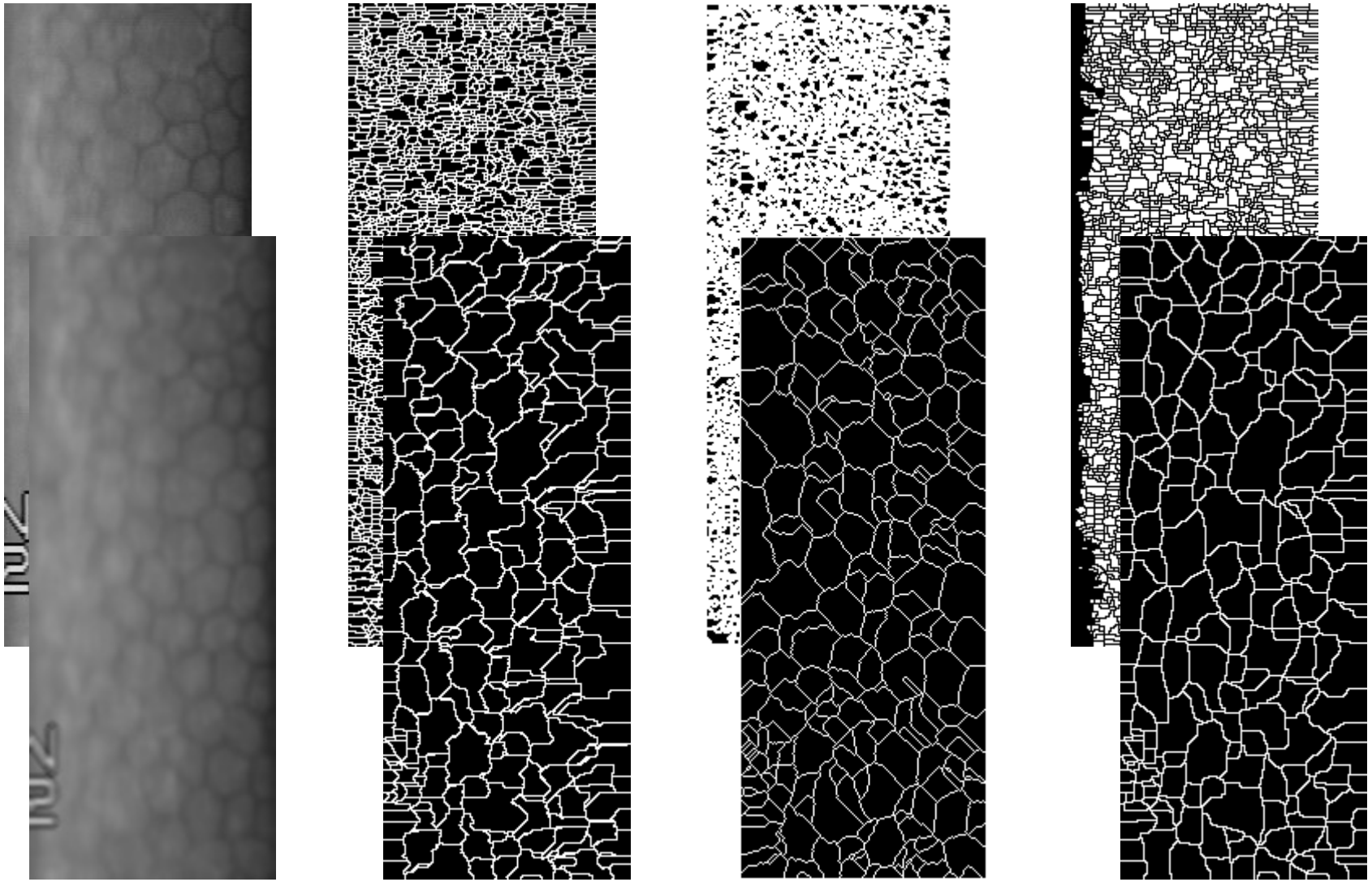
after smoothing

Corneal endothelium images

ImageJ

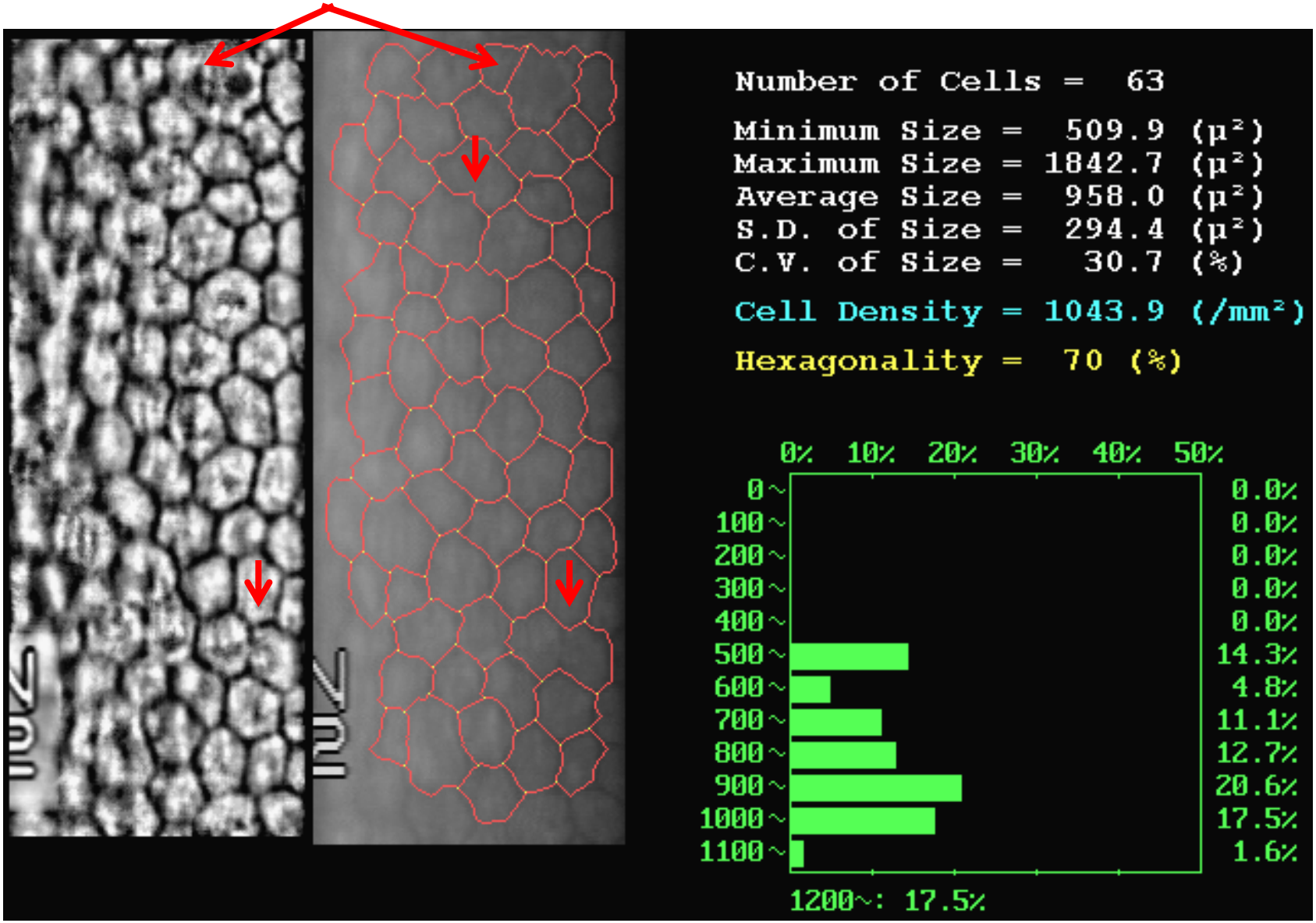
ImageJ - BIG

Matlab ☺

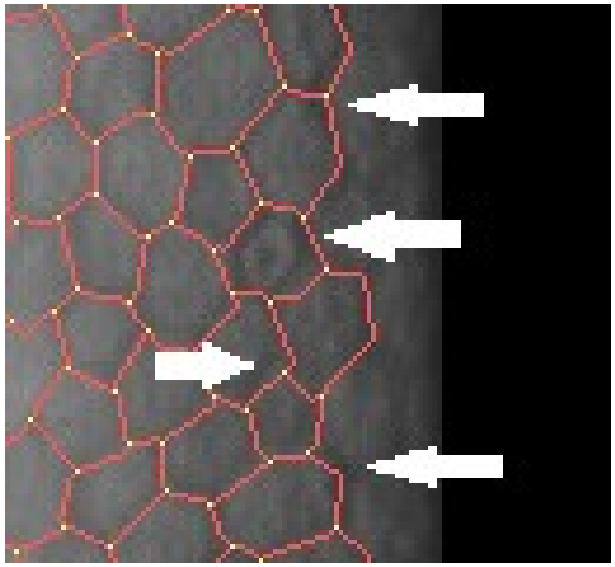


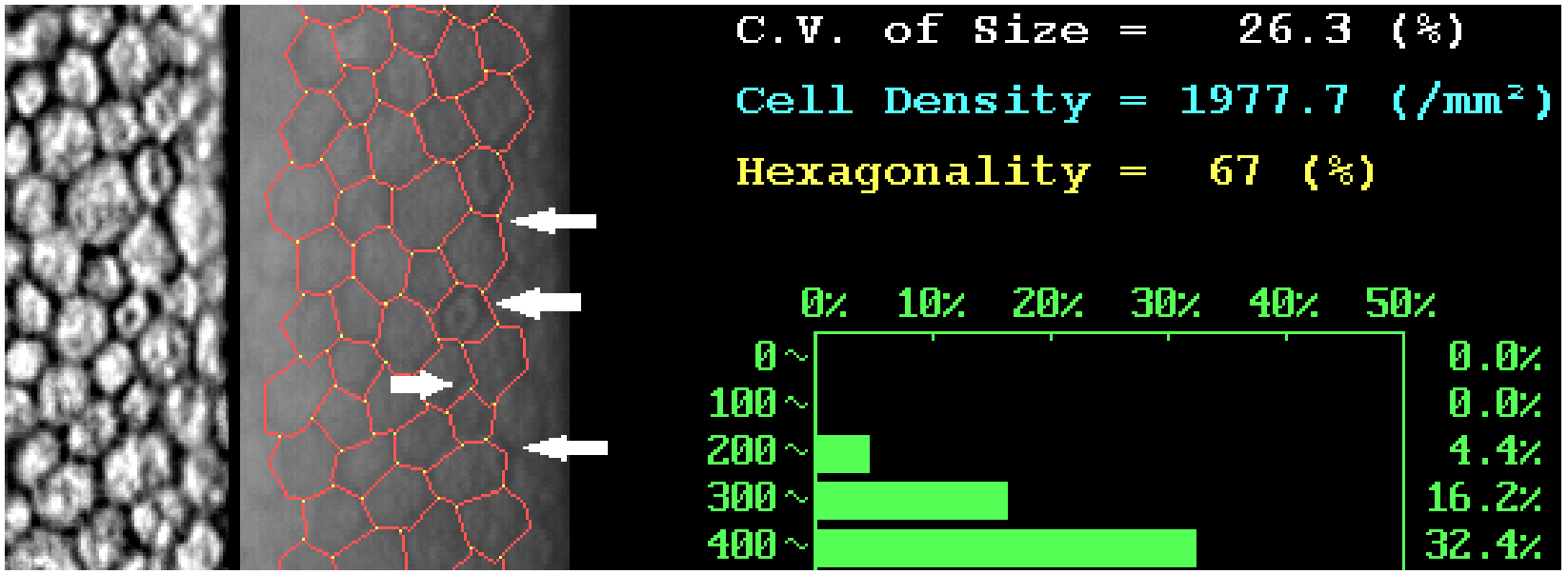
After smoothing

Corneal endothelium images



Corneal endothelium images





Preprocessing

$$S = \frac{\sum_{i=1}^{x} \sum_{j=1}^{y} Pin_{ij}}{xy} \tag{1}$$

$$Sw_j = \sum_{i=1}^{x} Pin_{ij} \tag{2}$$

$$Sk_j = \sum_{j=1}^{y} Pin_{ij} \tag{3}$$

$$Pout_{ij} = \begin{Bmatrix} Pin_{ij} + S - Sw_j & for\ Sw_j \neq S \\ Pin_{ij} & for\ Sw_j = S \end{Bmatrix} \tag{4}$$

$$Pout_{ij} = \begin{Bmatrix} Pin_{ij} + S - Sk_i & for\ Sk_i \neq S \\ Pin_{ij} & for\ Sk_i = S \end{Bmatrix} \tag{5}$$

Preprocessing

$$Pout_{ij} = \begin{cases} Pin_{ij} + S - Sw_j & for\ Sw_j > S \\ Pin_{ij} & for\ Sw_j = S \\ \frac{Pin_{ij} S}{Sw_j} & for\ Sw_j < S \end{cases} \quad (6)$$

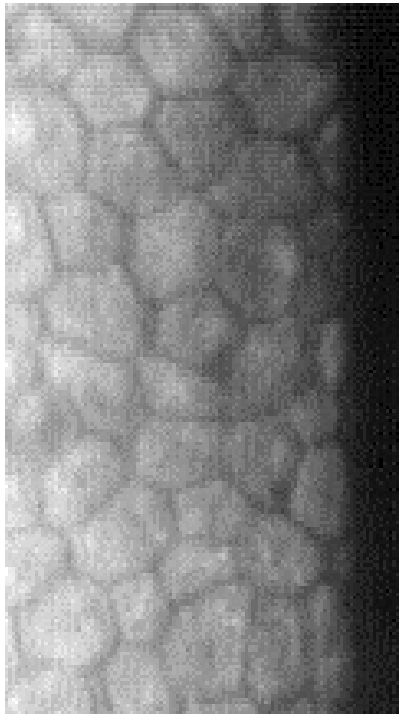
$$Pout_{ij} = \begin{cases} Pin_{ij} + S - Sk_i & for\ Sk_i > S \\ Pin_{ij} & for\ Sk_i = S \\ \frac{Pin_{ij} S}{Sk_i} & for\ Sk_i < S \end{cases} \quad (7)$$

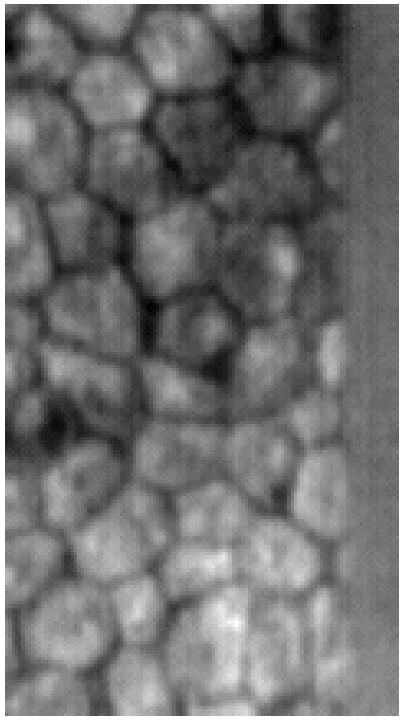
$$Pout_{ij} = \begin{cases} Pin_{ij} + S - Sk_i & for\ Sk_i > S \\ Pin_{ij} & for\ Sk_i = S \\ \frac{Pin_{ij} S}{Sk_i} & for\ Sk_i < S \end{cases} \quad (8)$$

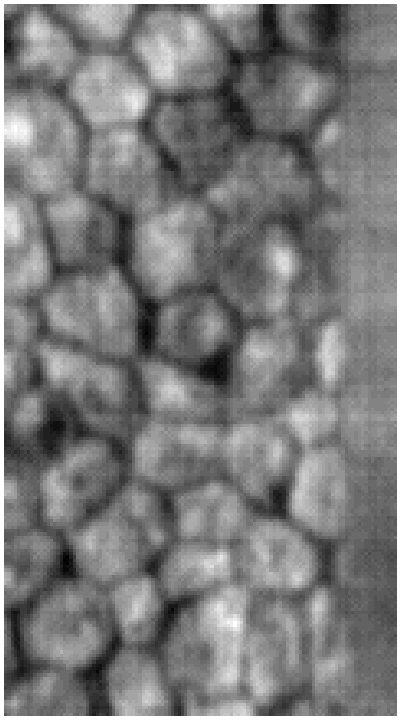
$$Pout_{ij} = Pin_{ij} + (S - Sr_{ij}) \quad (9)$$

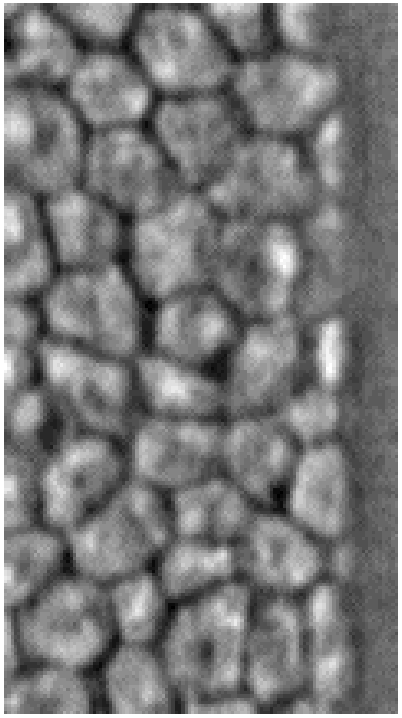
dodanie różnicy średniej obrazu (S) i średniej w kole o promieniu r dla piksela i,j (Srij) (9)

Preprocessing





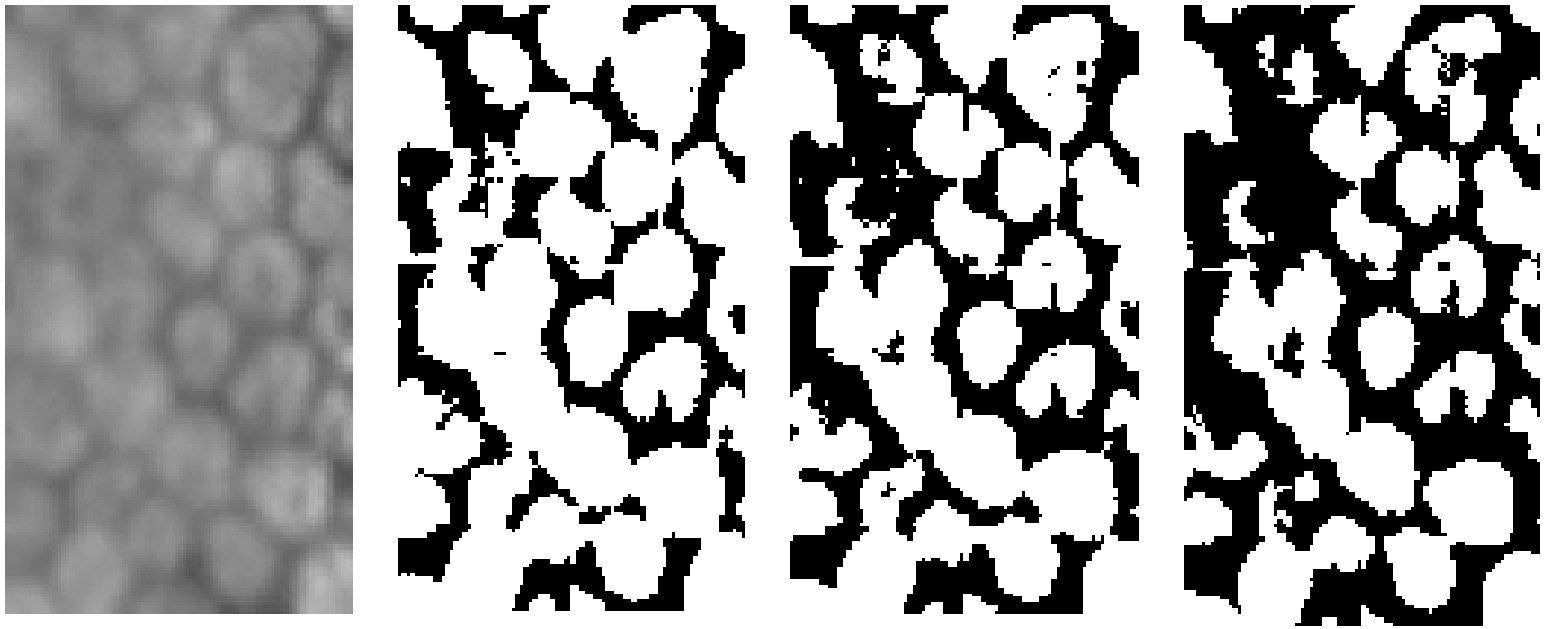




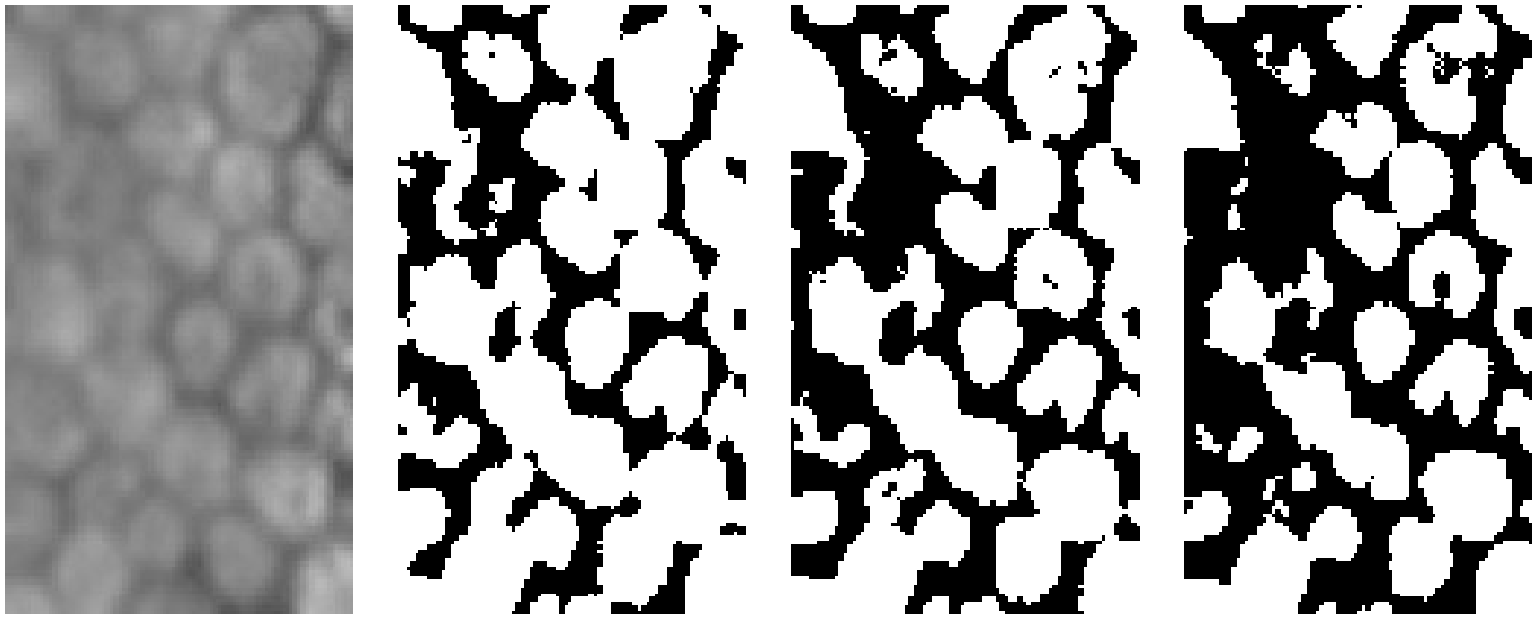
dodanie różnicy średniej obrazu
i średniej w kole (9)

r=10

Adaptive binarization

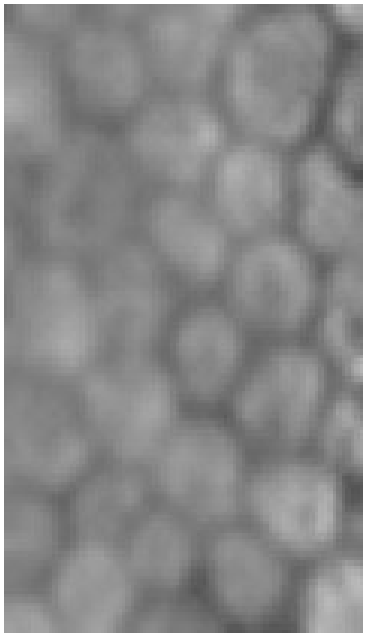


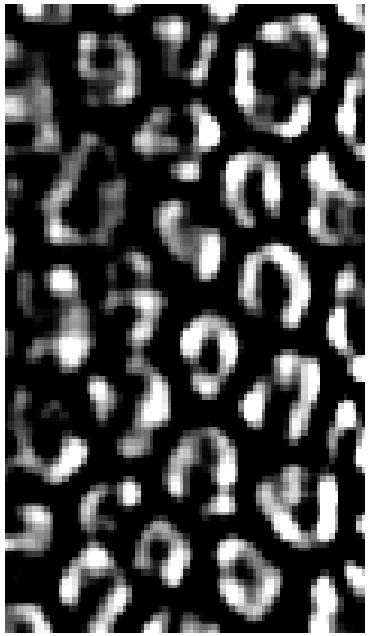
Top-Hat



Single 9x9 mask

$$\begin{bmatrix} -3 & -3 & -2 & -2 & -2 & -2 & -2 & -3 & -3 \\ -3 & -2 & -1 & -1 & -1 & -1 & -1 & -2 & -3 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -2 \\ -2 & -1 & 0 & 6 & 12 & 6 & 0 & -1 & -2 \\ -2 & -1 & 0 & 12 & 32 & 12 & 0 & -1 & -2 \\ -2 & -1 & 0 & 6 & 12 & 6 & 0 & -1 & -2 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -2 \\ -3 & -2 & -1 & -1 & -1 & -1 & -1 & -2 & -3 \\ -3 & -3 & -2 & -2 & -2 & -2 & -2 & -3 & -3 \end{bmatrix}$$





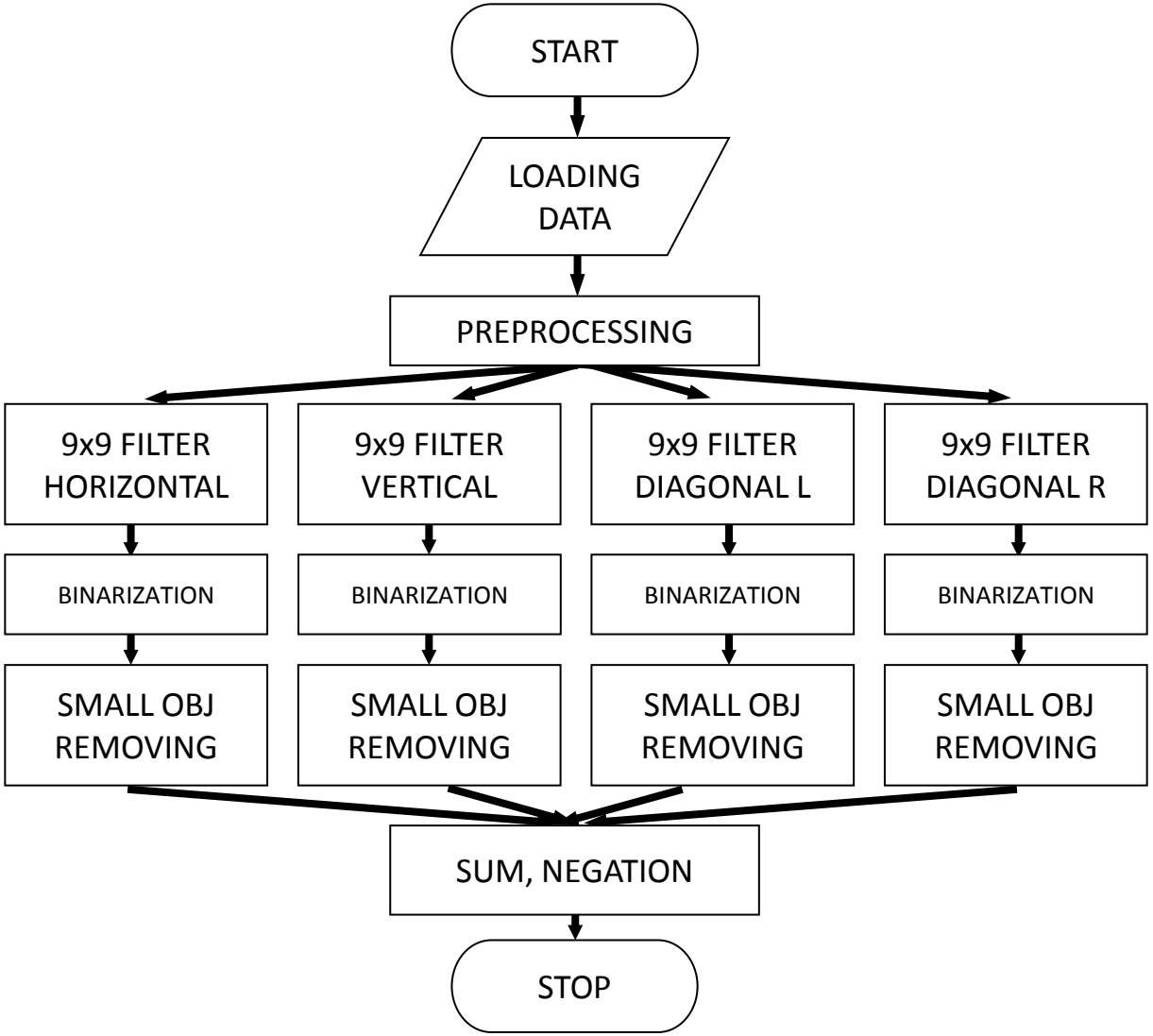
KH Algorithm

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 22 & -2 & -4 & -8 & -16 & -8 & -4 & -2 & 22 \\ 22 & -2 & -4 & -8 & -16 & -8 & -4 & -2 & 22 \\ 22 & -2 & -4 & -8 & -16 & -8 & -4 & -2 & 22 \\ 22 & -2 & -4 & -8 & -16 & -8 & -4 & -2 & 22 \\ 22 & -2 & -4 & -8 & -16 & -8 & -4 & -2 & 22 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 22 & 22 & 22 & 22 & 22 & 0 & 0 \\ 0 & 0 & -2 & -2 & -2 & -2 & -2 & 0 & 0 \\ 0 & 0 & -4 & -4 & -4 & -4 & -4 & 0 & 0 \\ 0 & 0 & -8 & -8 & -8 & -8 & -8 & 0 & 0 \\ 0 & 0 & -16 & -16 & -16 & -16 & -16 & 0 & 0 \\ 0 & 0 & -8 & -8 & -8 & -8 & -8 & 0 & 0 \\ 0 & 0 & -4 & -4 & -4 & -4 & -4 & 0 & 0 \\ 0 & 0 & -2 & -2 & -2 & -2 & -2 & 0 & 0 \\ 0 & 0 & 22 & 22 & 22 & 22 & 22 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 22 & 22 & 22 & 0 & 0 & 0 & 0 & 0 & 0 \\ 22 & -2 & -2 & -2 & 0 & 0 & 0 & 0 & 0 \\ 22 & -2 & -4 & -4 & -4 & -8 & -16 & 0 & 0 \\ 0 & -2 & -4 & -8 & -8 & -16 & -8 & 0 & 0 \\ 0 & 0 & -4 & -8 & -16 & -8 & -4 & 0 & 0 \\ 0 & 0 & -8 & -16 & -8 & -8 & -4 & -2 & 0 \\ 0 & 0 & -16 & -8 & -4 & -4 & -4 & -2 & 22 \\ 0 & 0 & 0 & 0 & 0 & -2 & -2 & -2 & 22 \\ 0 & 0 & 0 & 0 & 0 & 0 & 22 & 22 & 22 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 22 & 22 & 22 \\ 0 & 0 & 0 & 0 & 0 & -2 & -2 & -2 & 22 \\ 0 & 0 & -16 & -8 & -4 & -4 & -4 & -2 & 22 \\ 0 & 0 & -8 & -16 & -8 & -8 & -4 & -2 & 0 \\ 0 & 0 & -4 & -8 & -16 & -8 & -4 & 0 & 0 \\ 0 & -2 & -4 & -8 & -8 & -16 & -8 & 0 & 0 \\ 22 & -2 & -4 & -4 & -4 & -8 & -16 & 0 & 0 \\ 22 & -2 & -2 & -2 & 0 & 0 & 0 & 0 & 0 \\ 22 & 22 & 22 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

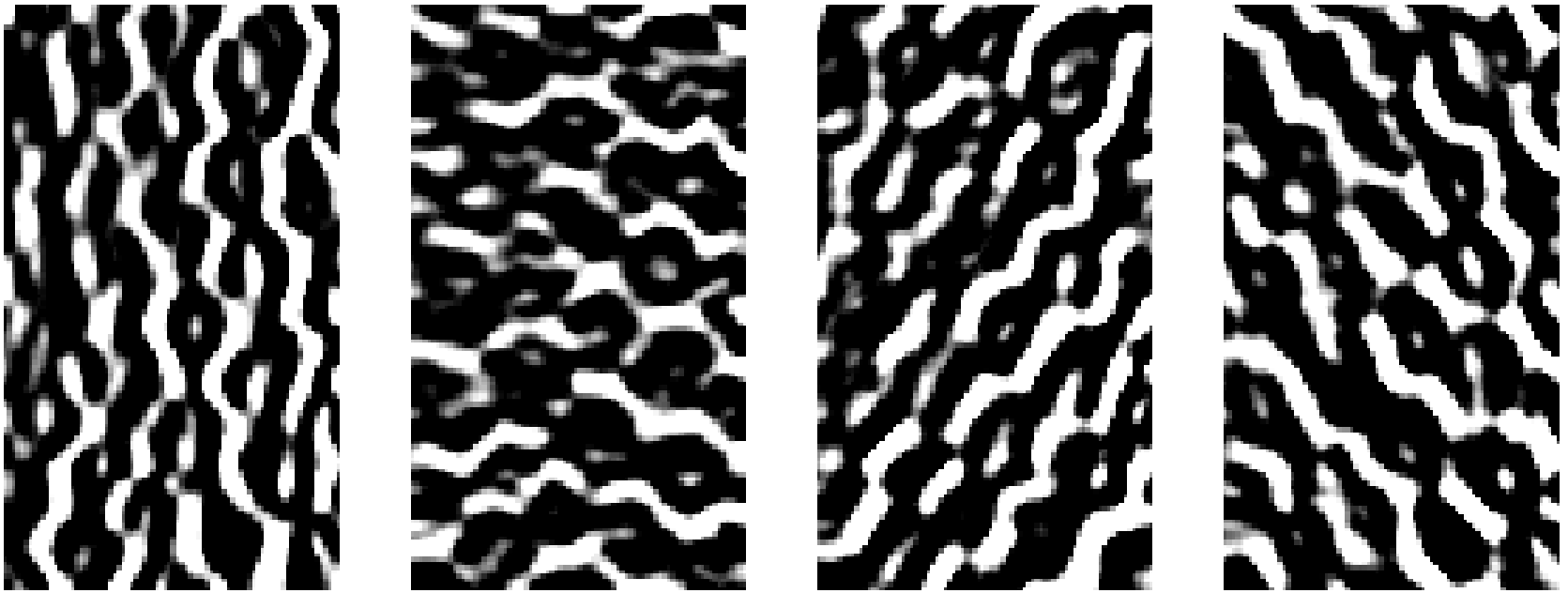
K. Habrat, M. Habrat, J. Gronkowska-Serafin, A. Piorkowski: Cell Detection In Corneal Endothelial Images Using Directional Filters. Springer 2016, AISC Vol. 389, pp 113–123.

KH Algorithm



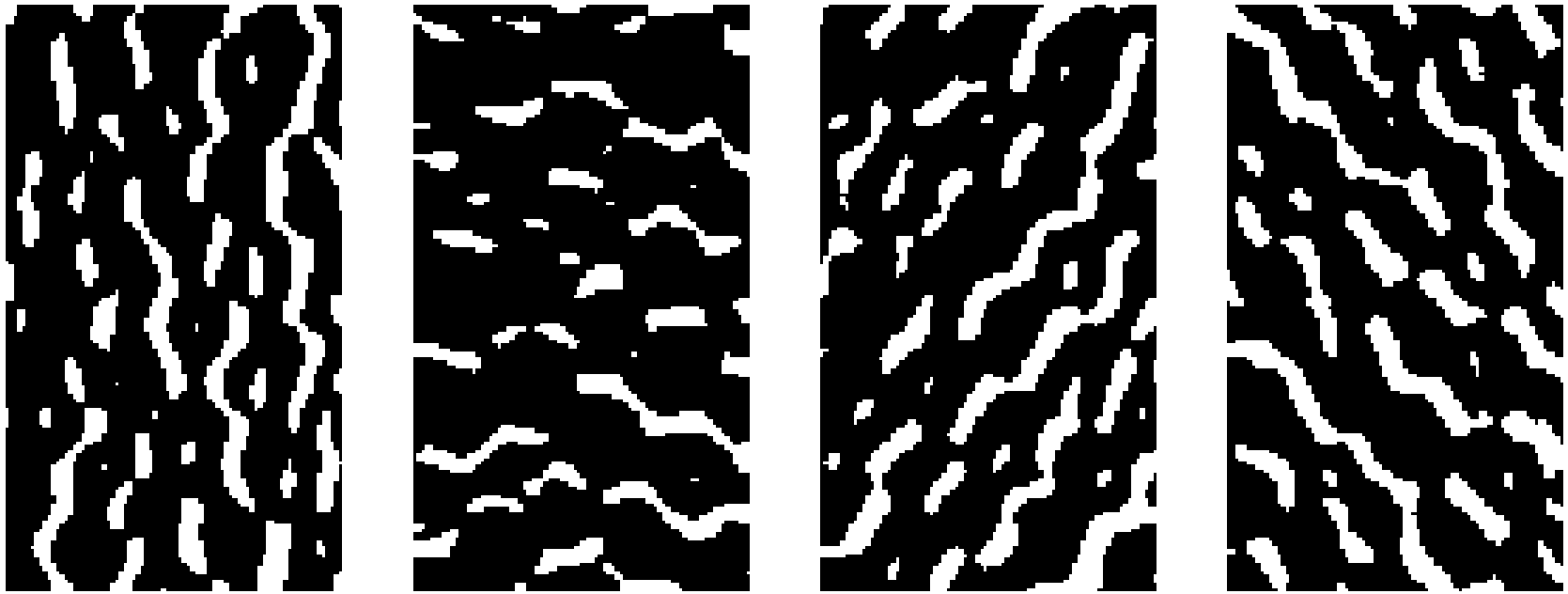
KH Algorithm

… using masks



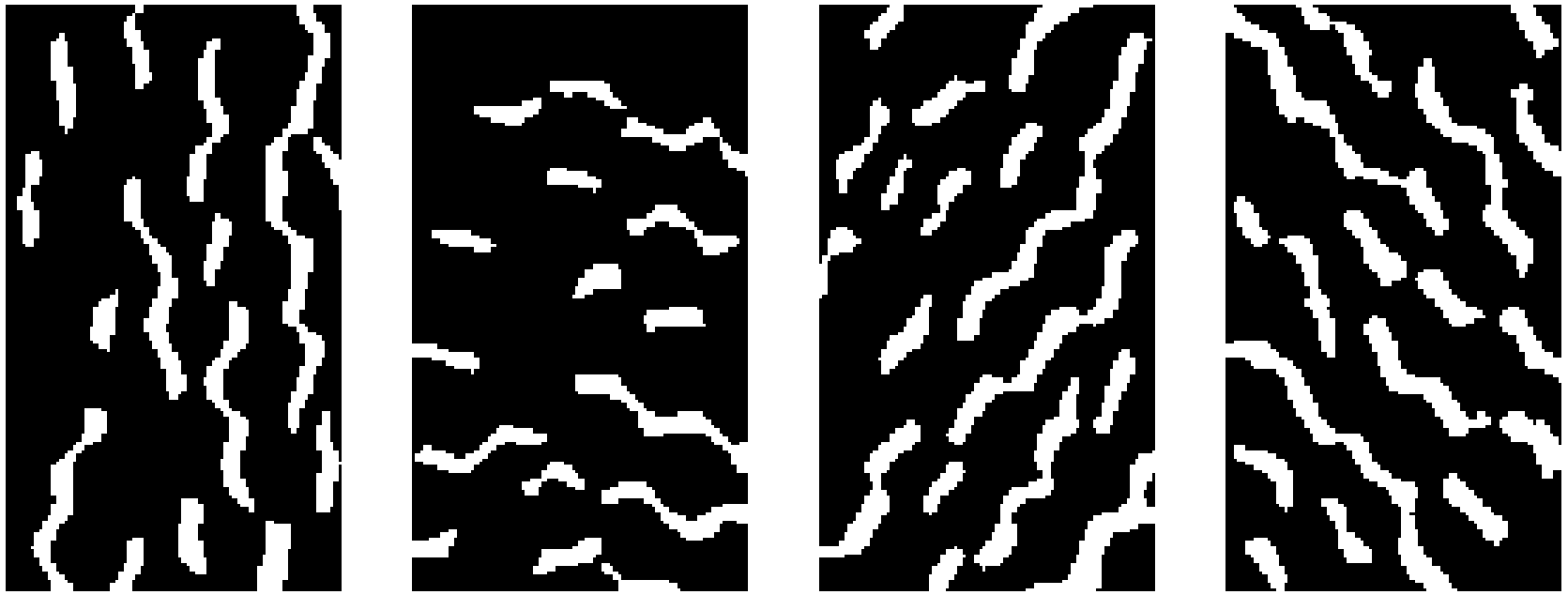
KH Algorithm

… and next the binarizaton (~50% of range)



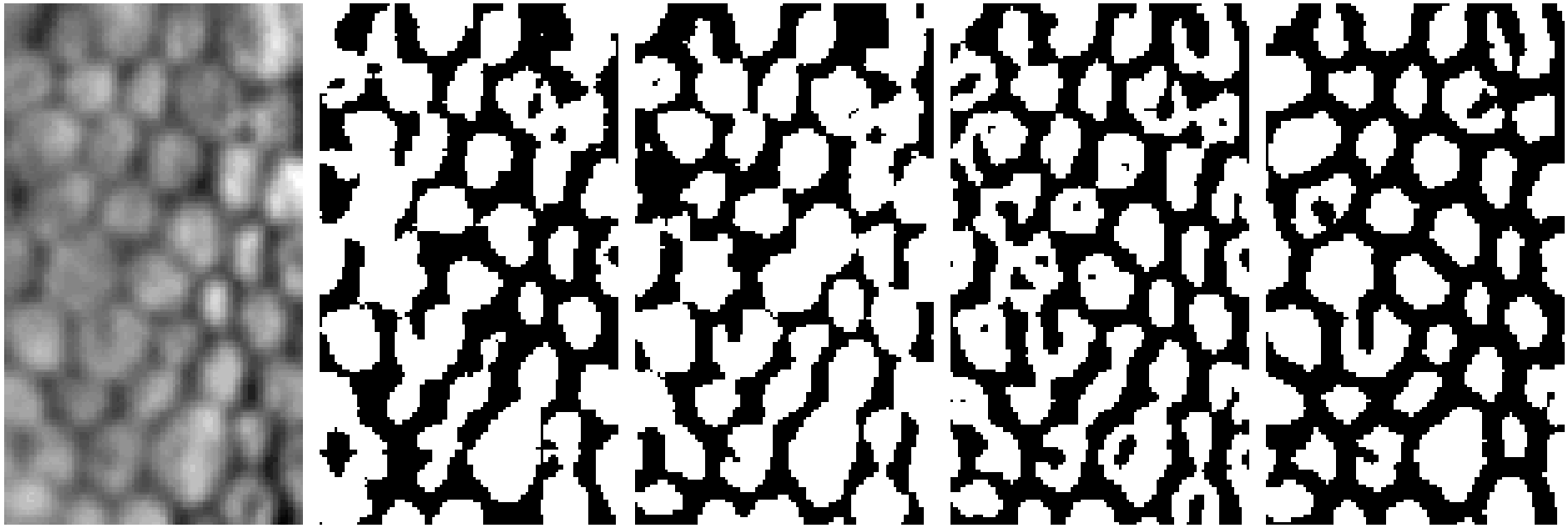
KH Algorithm

… small object removing at the end (40px)

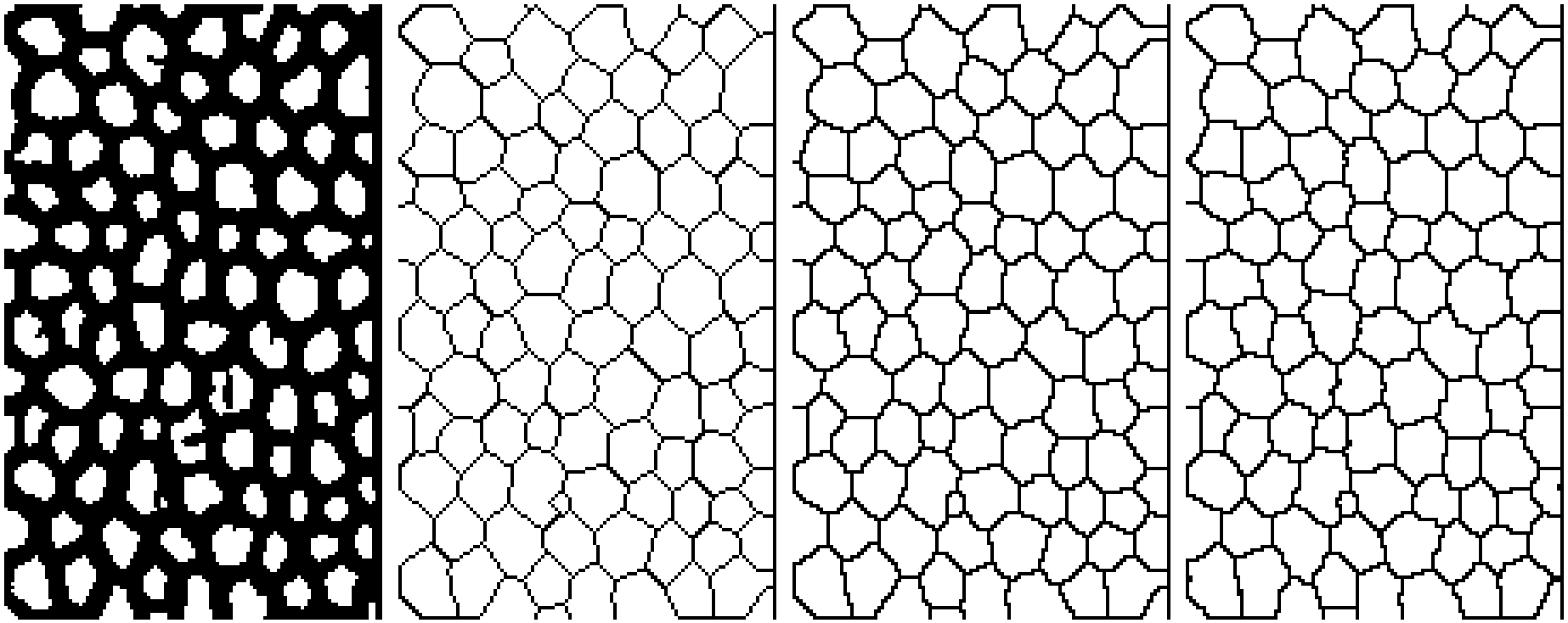


KH Algorithm

org, ench. | adaptive bin | Top-Hat | Single Mask | KH Alg.



THINNING



KH bin,
inverted

thinning
using mask A

thinning
using mask B

thinning
using mask C