

## Wprowadzenie do Image Processing Toolbox

### 1 Operacje I/O

Do wczytywania obrazów w MatLABie służy polecenie `imread('nazwa_pliku')` np. `a=imread('cameraman.tif')` oraz `[map, legend]=imread(obraz);`

Do zapisywania tablic do plików graficznych służy polecenie `imwrite(zmienna, 'nazwa pliku', opcje)`. Opcje są zależne od wybranego formatu zapisu obrazu. Szczegółowe informacje o możliwych opcjach zawarte są w pomocy tej funkcji.

Do wyświetlania służy polecenie `imshow(zmienna, opcje)`, np. `imshow(a, 'InitialMagnification', 'fit')` oraz: `imagesc(x,y, obraz);` - dla obrazów poza skalą 0-1.

Dodatkowym poleceniem jest `imtool(obraz)`, które oprócz wyświetlania obrazu, udostępnia wiele narzędzi: pixel region, distance, image intensity, adjust contrast, choose colormap.

Dostępne formaty plików to bmp, cur, gif, hdf, ico, jpg, pbm, pcx, pgm, png, pnm, ppm, ras, tif, xwd.

### 2 Formaty przechowywania obrazu

W pakiecie MatLAB obrazy są przechowywane na kilka sposobów:

- BIN - macierz binarna (logiczna), przyjmująca wartości true / false
- GRAY - macierz poziomów szarości. Wartości mogą być typu double (0-1), uint8 (0-255) lub uint16.
- RGB - potrójna tablica intensywności (3 tablice typu GRAY)
- IND - tablica indeksowa (składa się z 2 części: mapy i legendy), gdzie legenda jest tablicą o wymiarze Nx3 (gdzie N - ilość dostępnych kolorów), a mapa jest tablicą z numerami kolorów (numerami odpowiednich wersów w legendzie).

### 3 Formaty kodowania koloru

Istnieje wiele sposobów kodowania koloru. Oprócz najpopularniejszego RGB, bazującego na ludzkim sposobie postrzegania barw czy używanym w drukarkach CMYK, istnieją również specyficzne formaty mające na ogół bardzo wąskie zastosowanie. Generalnie podzielić je można ze względu na ilość tablic (kanałów) służących do przechowywania pojedynczego koloru.

1. Pojedyncza tablica: Gray
2. Potrójna tablica:
  - RGB (Red, Green, Blue)
  - YCbCr, zwane YUV (Luminance, Chrominance)
  - H S V (Hue, Saturation, Value) - oparte o stożek: H=0:360; S,V=0:100.
  - CMY
  - $L^*a^*b^*$  - (Luminance, a i b)
  - XYZ
  - HLS (Hue, Saturation, Lightness)

3. Poczwórna tablica: CMYK (Cyan, Magenta, Yellow, Key (black))
4. n-kanalowe.

Przykładowe wartości kolorów przedstawia tab.1.

**Tabela 1:** Wartości przykładowych kolorów w wybranych kodowaniach koloru (UWAGA: kolory XYZ, CMY, CMYK zostały określone na podstawie RGB przy użyciu <http://www.easycrgb.com/index.php?X=CALC>, natomiast HSV, YUV i YIC na podstawie: <http://web.forret.com/tools/color.asp>)

format	czarny	biały	czerwony	zielony	niebieski
RGB	0, 0, 0	1, 1, 1	1, 0, 0	0, 1, 0	0, 0, 1
XYZ	0, 0, 0	95.05, 100, 108.9	41.24, 21.26, 1.93	35.76, 71.52, 11.92	18.05, 7.22, 95.05
CMY	1, 1, 1	0, 0, 0	0, 1, 1	1, 0, 1	1, 1, 0
CMYK	0, 0, 0, 1	0, 0, 0, 0	0, 100, 100, 0	100, 0, 100, 0	100, 100, 0, 0
HSV	0, 0, 0	0, 0, 100	0, 100, 100	120, 100, 100	240, 100, 100
HSL	0, 0, 0	0, 0, 100	0, 100, 50	120, 100, 50	240, 100, 50
YUV	0, 0, 0	100, 0, 0	29.9, -14.7, 61.5	58.7, -28.9, -51.5	11.4, 43.6, -10
YIC	0, 0, 0	100, 0, -62.2	29.0, 59.6, 21.1	58.7, -27.4, -52.3	11.4, -32.1, -31.1

Konwersji pomiędzy tymi formatami i typami używa się przy wykorzystaniu funkcji o nazwie `typ2typ`, np.: `rgb2gray()` czy `ycbcr2rgb()`.

Dla przykładu, przekształcenie RGB->YIQ polega na macierzowym mnożeniu macierzy przekształcenia przez wektor RGB. Macierz przekształcenia przedstawiano jest poniżej. Format YIQ jest powszechnie wykorzystywany przy analizie barwionych szlifów mikroskopowych

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & -0.311 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Z kolei przekształcenie RGB->CMY i odwrotne, polega na odjęciu od największej dopuszczalnej wartości danej palety:

CMY = 255 - RGB (dla uint8) i RGB = 1 - CMY (dla double).

## 4 Palety barw

Do wyświetlania palet służy polecenie `rgbplot(nazwa(dynamika))`. W MatLABie dostępne są następujące palety:

- hsv (Hue, Saturation, Value)
- Jet - przestawione hsv
- Hot - ciepłe kolory: czarny, czerwony, żółty, biały
- Cool - zimne kolory: odcienie błękitu i purpury
- Gray - odcienie szarości
- Bone - odcienie niebieskiego
- Cooper - odcienie miedzi
- Pink - odcienie różu
- Prism - kolory: zielony, żółty, czerwony, fiolet, niebieski
- flag - kolory: czerwony, biały, niebieski, czarny
- spring, summer, autumn, winter

Narzucanie palety barw na figurę odbywa się przy użyciu polecenie `colormap( paleta)`. Narzucenie palety odbywa się po wyświetleniu obrazu i dotyczy wszystkich obrazów na danej figurze.

## 5 Informacje o obrazie

Do wyświetlania informacji o pliku obrazu: `imfinfo('nazwa_pliku');`

Innych informacji o obrazie udziela funkcja `regionprops(obraz, 'all')`. Udziela ona o powierzchni, położeniu środka ciężkości, etc.

Do wycinania konkretnego fragmentu służy funkcja `imcrop(obraz, rect)`, gdzie `rect` = `[xmin, ymin, szerokość, wysokość]`. Nie podanie opcji `rect` powoduje, wyznaczenie obszaru wycinku poprzez zaznaczenie urządzeniem wskazującym (np. kursorem myszki).

Do uzyskiwania informacji o kolorze pikseli znajdujących się pod linią (lub łamaną) służy polecenie `linia=improfile(obraz, [x1 x2], [y1 y2]);`. Do naniesienia owej linii na wyświetlany obraz służy polecenie `line([x1 x2], [y1 y2], 'color', [R G B]);`. RGB jest kodowane w wartościach typu `double`. Do uzyskiwania informacji o wartościach intensywności konkretnego piksela służy polecenie `piksel=impixel(obraz, x,y);` x - nr kolumny, y - nr wiersza. Dany piksel można zaznaczyć na wyświetlanym obrazie np. poleceniem `text(x,y, '*', 'color', [R G B])`. Należy pamiętać o zmianie kolejności numerowania. Polecenie `impixel(obraz, x, y)` jest tożsame z komendą `obraz(y, x)` w wierszu poleceń.

## 6 Rozdzielczość przestrzenna

Rozdzielczość przestrzenna definiowana jest na wiele sposobów. Jedną z najczęściej spotykanych definicji stanowi, że jest to rozmiar powierzchni jaką zajmuje dany piksel (wielkość terenu). W przypadku wydruków i obrazów często spotykaną miarą rozdzielczości przestrzennej jest DPI (dot per inch) czyli ilość punktów (pikseli) na 1 cal wydruku lub skanu.

Do zmiany rozmiaru (a co za tym idzie rozdzielczości) służy polecenie `imresize(obraz, [rozmiar], 'metoda');`. W MatLABie istnieją 3 metody interpolacji:

- 'nearest' - najbliższego sąsiada,
- 'bilinear' - interpolacja dwuliniowa,
- 'bicubic' - interpolacja kubiczna (dwukwadratowa).

## 7 Rozdzielczość barwna

Rozdzielczość poziomów szarości dla obrazów monochromatycznych  $R_L$  definiowana jest jako (Wróbel & Koprowski):

$$\frac{1}{R_L} = \frac{l_w - l_n}{P} \quad (1)$$

gdzie:

$l_w$  i  $l_n$  - wysoki i niski poziom szarości

P - liczba naturalna  $P \in \{0, 1, 2, \dots, 2^b - 2, 2^b - 1\}$

b - liczba bitów służących do reprezentacji danego poziomu szarości

Do zmiany dynamiki obrazu (ilości wyświetlanych odcieni) służy polecenie: `imapprox( mapa, legenda, dynamika)`. Na wejściu i wyjściu używane są obrazy indeksowane (konwersja `rgb2ind, gray2ind`).

## Przekształcenia punktowe i geometryczne

### 1 Przekształcenia punktowe

Przekształcenia punktowe (bezkontekstowe) są to przekształcenia dotyczące stopnia szarości lub nasycenia barwy dla każdego punktu oddzielnie, dla których nie mają wpływu wartości w punktach sąsiednich.

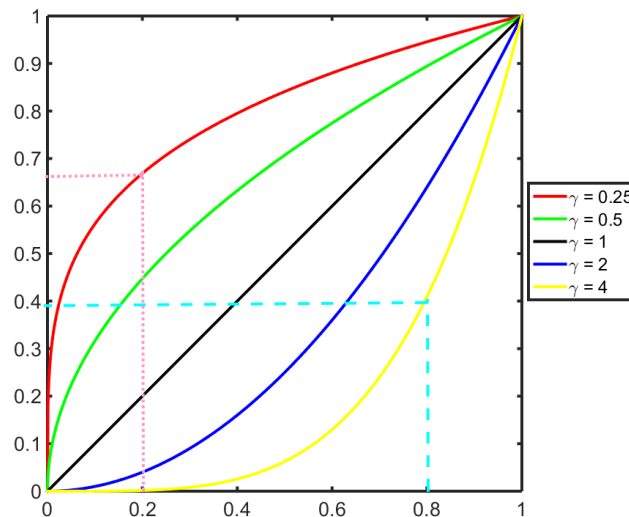
#### 1.1 Liniowe przekształcenia obrazu

Liniowe przekształcenie polega na zmianie wartości piksela przy użyciu przekształcenia liniowego. Zaliczamy do nich między innymi operacje dodawania, negacji czy mnożenia przez pewną liczbę. Służą one np. do rozjaśniania obrazu. Należy pamiętać, że przy stosowaniu kodowania `uint8` dozwolone wartości są z przedziału 0-255, a dla kodowania `double` dozwolone są wartości z przedziału 0-1.

#### 1.2 Nieliniowe przekształcenia obrazu

Do nieliniowych przekształceń zaliczamy np. pierwiastkowanie, potęgowanie logarytmowanie. W niektórych przypadkach jest to znacznie skuteczniejsze niż liniowe korekcje. Szczególne znaczenie ma korekcja gamma  $\gamma$  dana wzorem eq. 1. Idee działania współczynnika  $\gamma$  na obraz przedstawia Rys. 1.

$$\text{Nowy}(m, n) = \text{Stary}(m, n)^\gamma \quad (1)$$



**Rysunek 1:** Idea działania korekcji  $\gamma$ . Dla  $\gamma < 1$  zwiększony jest kontrast dla "ciemnej" części obrazu kosztem "jasnej" części. Dla  $\gamma > 1$  efekt jest odwrotny: zyskujemy kontrast w części "jasnej" kosztem "ciemnej".

### 1.3 Normalizacja

Normalizacja ma na celu rozciągnięcie zakresu tonalnego obrazu. Jeżeli intensywność punktów na obrazie zajmuje tylko część dozwolonej skali, to możliwe jest rozciągnięcie intensywności tak, by zajmowała pełną skalę. Stosowana jest normalizacja zarówno w wymiarze globalnym, jak i lokalnym. Jeżeli przez  $M$  oznaczamy minimum w normalizowanym obszarze, przez  $N$  maksimum, a przez  $L$  i  $U$  odpowiednio minimum i maksimum skali, to intensywność punktu wynosi:

$$Nowy(m, n) = \left( \frac{L - U}{M - N} \cdot (Stary(m, n) - M) \right) + L \quad (2)$$

Obraz metodą globalną normalizuje się przy użyciu funkcji:

`imadjust(obraz, [low_in; high_in], [low_out; high_out], gamma).`

### 1.4 Binarizacja

Binarizacja to zamiana dowolnego obrazu na obraz logiczny. Stosuje się kilka rodzajów binaryzacji:

- z dolnym progiem: wszystko poniżej progu: 0, powyżej 1

$$\mathbf{L}(\mathbf{m}, \mathbf{n}) = \begin{cases} 0 & \text{gdy } L(m, n) \leq prog \\ 1 & \text{gdy } L(m, n) > prog \end{cases}$$

- z górnym progiem: negatyw binaryzacji z dolnym progiem

$$\mathbf{L}(\mathbf{m}, \mathbf{n}) = \begin{cases} 0 & \text{gdy } L(m, n) > prog \\ 1 & \text{gdy } L(m, n) \leq prog \end{cases}$$

- z dwoma progami: przypomina funkcję bramkową lub jej negatyw
- wielokryterialna: kilka progów i tylko niektóre przedziały mają wartość prawda / fałsz.
- z histerezą: wymaga podania 2 wartości progowych ( $prog1 < prog2$ ), wartość piksela  $L(m, n)$  dana jest wzorem:

$$\mathbf{L}(\mathbf{m}, \mathbf{n}) = \begin{cases} 0 & \text{gdy } L(m, n) \leq prog1 \\ s & \text{gdy } prog1 < L(m, n) \leq prog2 \\ 1 & \text{gdy } L(m, n) > prog2 \end{cases}$$

$s$  - wartość sąsiadujących punktów.

- automatyczne progowanie metodą maksymalnej entropii

Wartość entropii  $H$  dla obrazu  $\mathbf{A}$  definiowana jest poprzez eq. 3.

$$H = - \sum_{k=0}^{N_{kolor}} (p(k) \cdot \ln p(k)) \quad (3)$$

gdzie  $p(k)$  - prawdopodobieństwo wystąpienia intensywności  $k$  w obrazie  $\mathbf{A}$ ,  $N_{kolor}$  - maksymalna wartość intensywności występująca w obrazie  $\mathbf{A}$ .

Binarizacja metodą maksymalnej entropii polega na znalezieniu takiej wartości progu  $k$ , dzielącej obraz na tło (piksele o intensywności mniejszej lub równej  $k$ )

i obiekty (o intensywności większej niż  $k$ ), która maksymalizuje funkcję  $H(k)$  (eq. 4). Metoda ta wymaga, by obiekty i tło miały w przybliżeniu podobną powierzchnię.

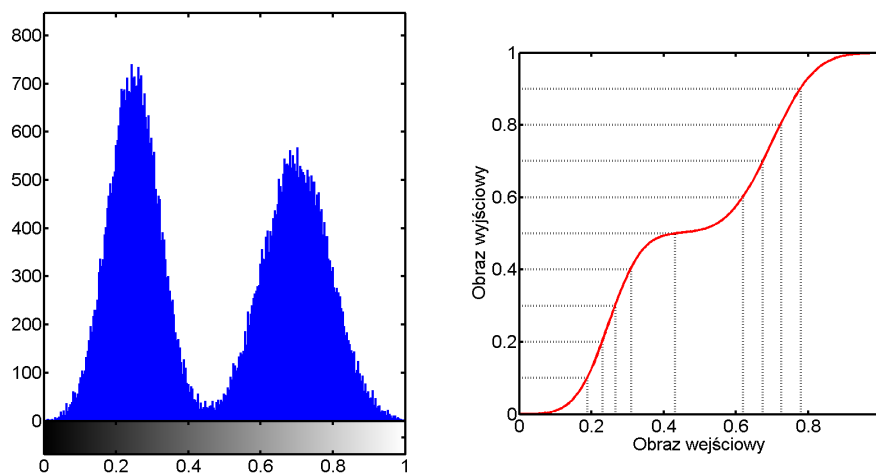
$$H(k) = -\frac{1}{N_k(tlo)} \sum_{i=0}^k (p(i) \cdot \ln p(i)) - \frac{1}{N_k(obiekt)} \sum_{i=k+1}^{N_{kolor}} (p(i) \cdot \ln p(i)) \quad (4)$$

gdzie  $N_k(tlo)$  - ilość elementów tła, a  $N_k(obiekt)$  - ilość pikseli obiektów uzyskanych przy binaryzacji z progiem  $k$ ,  $i$  - intensywność.

## 2 Wyrównywanie histogramu

Histogram wykreśla się poleceniem `imhist(obraz)`. Histogram jest to wykres słupkowy określający ile punktów przybiera daną wartość (intensywność). Wyrównywanie histogramu polega na "upodobnieniu" go do histogramu dla rozkładu równomiernego (wszystkie intensywności mają jednakową częstotliwość występowania) (Rys. 2). Do wyrównywania histogramu służy polecenie `histeq(obraz, ilość_klas)`. Czasami stosuje się wyrównywanie lokalne, tzn. dzieli się obraz na kilka części i każdą wyrównuje oddzielnie. Do tego stosuje się funkcję `adapthisteq(obraz, parametry)`. Do najważniejszych parametrów tego przekształcenia zaliczamy:

- **NumTiles** - Wielkość elementu, dla którego wyrównujemy histogram
- **ClipLimits** - skalarna wielkość decydująca o kontraście (0,1)
- **NBins** - Ilość klas histogramu
- **Range** - zakres wyrównywanych wartości
- **Distribution** - rodzaj histogramu, do którego będzie normowany obraz: 'uniform' (płaski), 'Rayleigh' (dzwonkowy) i 'Exponential' (krzywa eksponenty).
- **Alpha** - parametr rozkładu histogramu, tylko dla Rayleigh i Exponential.



**Rysunek 2:** Idea wyrównywania histogramu obrazu. 1) Liczymy histogram; 2) Liczymy kumulantę (czerwona linia) i dzielimy ją przez jej maksymalną wartość; 3) Dzielimy Oś OY na wykresie kumulanty na "n" równych odcinków (klas histogramu wyjściowego); 4) Szukamy wartości rzutów wartości poszczególnych progów na Oś OX poprzez kumulantę; 5) Wszystkie wartości z OX z ograniczone dwoma rzutami zamieniamy na odpowiadającą im wartość OY.

### 3 Przekształcenia geometryczne

Do przekształceń geometrycznych zaliczamy między innymi: przesunięcie, obrót, zniekształcenie, odbicia symetryczne, powielanie fragmentów. Mają one wpływ na kształt, wielkość lub wygląd obrazu.

#### 3.1 Przesunięcie o wektor

Możliwe są przesunięcia o wektor z dodaniem nowej powierzchni (zwiększenie wymiarów obrazu wyjściowego o wektor) lub przesunięcie w sposób cykliczny z użyciem funkcji `circshift(obraz, wektor)`.

#### 3.2 Obrót

Obraz można obracać przy użyciu funkcji `imrotate(obraz, kąt w stopniach, opcje)`. Pierwszą opcją są metody interpolacji nowych danych:

- najbliższego sąsiada (`'nearest'`);
- dwuliniowa (`'bilinear'`);
- bikubiczna (`'bicubic'`).

Drugą opcją jest rozmiar wyjściowy obrazu:

- `'crop'`: Obraz wyjściowy ma taki sam rozmiar jak wejściowy. Rogi są obcięte, wolne miejsca wypełnione zerami.
- `'loose'`: Obraz wyjściowy na ogół jest powiększony tak, by zmieścił się cały obraz. Wolne miejsca wypełnione zerami.

Opcjami domyślnymi jest `'nearest'`, `'loose'`. Użycie kąta dodatniego powoduje obrót przeciwnie do kierunku wskazówek zegara.

Rotacja obrazu polega na wyliczeniu nowych współrzędnych danego piksela obrazu. Sprowadza się to do wymnożenia wektora współrzędnych przez macierz rotacji:

$$\begin{bmatrix} x^{obr} \\ y^{obr} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (5)$$

#### 3.3 Odbicia symetryczne

Odbicia symetryczne mogą być względem prostej pionowej (`fliplr (obraz)`) lub poziomej (`flipud (obraz)`). Odbicia te (a zwłaszcza odbicie poziome) mają zastosowanie podczas kalibracji (ang. registering) zdjęć (przypisania współrzędnych, np. geograficznych, do poszczególnych pikseli obrazu).

#### 3.4 Dodawanie wierszy, kolumn do obrazu

Dodawanie kolumn i wierszy do obrazu ma na celu głównie podczas filtracji w celu zmniejszenia efektów brzegowych. Służy do tego polecenie `padarray (obraz, rozmiar, metoda, kierunek)`;

Rozmiar w formie wektora `[x y]` mówi, ile wierszy `x` i ile kolumn `y` należy dodać.

Do metod zaliczamy:

- `'circular'` - doklejanie cykliczne

- 'replicate' - doklejanie poprzez kopiowanie skrajnych wiersów / kolumn
- 'symmetric' - odbicie symetryczne.

Istnieją 3 kierunki:

- 'pre': doklejanie z góry i z lewej;
- 'post': z dołu i z prawej;
- 'both': z obu jednocześnie.

### 3.5 Zmiana kształtu i wymiarów obrazu

Zmiana wymiarów obrazu odbywa się przy wykorzystaniu polecenia `reshape (obraz, [nowy rozmiar], 'opcja')`. Do poprawnego działania funkcji musi być spełniony warunek zachowania powierzchni:  $(x^{old} * y^{old} = x^{new} * y^{new})$

Do zmiany kształtu, bez konieczności spełnienia powyższego warunku może służyć polecenie `imresize(obraz, [nowy rozmiar], 'opcja')`.

Zmiana kształtu może odbywać się przy wykorzystaniu dwóch poleceń: `maketform (typ, parametry)` do tworzenia maski przekształcającej i `imtransform (obraz, forma)`; do wykonania samego przekształcenia.

W nowszych wersjach polecenie `maketform` zostało zastąpione poleceniami:

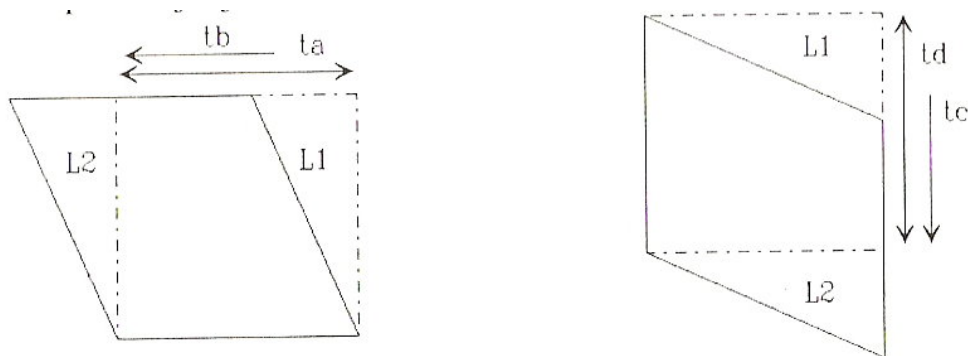
`affine2d([ macierz 3x3 ])` i `projective2d([ macierz 3x3 ])`. W przypadku afinicznym, ostatnia kolumna musi mieć wartości  $[0; 0; 1]$ , a dane w trzecim wierszu nie mają wpływu na obraz.

Polecenie `imtransform` zostało zastąpione natomiast poleceniem `imwarp(obraz, maska)`.

Wyróżniamy następujące typy transformacji:

- ('affine',  $[T_A T_B; T_C T_D; T_E T_F]$ ) - przekształcenie afiniczne: każde różnowartościowe przekształcenie geometryczne, które wszystkie proste zawarte w dziedzinie tego odwzorowania przekształca na proste (Rys. 3). Przekształcenie afiniczne dane jest wzorem:

$$[x_{new} y_{new}] = [x y 1] \cdot \mathbf{T} \quad (6)$$



**Rysunek 3:** Parametry przekształcenia afinicznego (źródło:Wróbel & Koprowski, 2004)

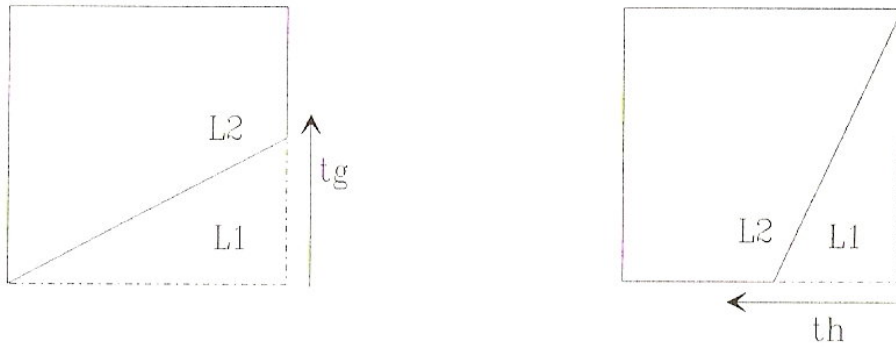
- ('projective',  $[T_A T_D T_G; T_B T_E T_H; T_C T_F T_I]$ ): takie przekształcenie, gdzie każde punkty leżące na jednej prostej przechodzą w punkty leżące na drugiej prostej (Rys. 4). Projektcja dane jest wzorem:

$$[x_{new} y_{new} z_{new}] = [x y w] \cdot \mathbf{T} \quad (7)$$



. W przypadku obrazów 2D, powyższe równania można zapisać w następujący sposób:

$$x_{new} = \frac{T_A \cdot x + T_B \cdot y + T_C}{T_G \cdot x + T_H \cdot y + T_I} \quad y_{new} = \frac{T_D \cdot x + T_E \cdot y + T_F}{T_G \cdot x + T_H \cdot y + T_I} \quad (8)$$



**Rysunek 4:** Parametry projekcji (źródło:Wróbel & Koprowski, 2004)

- 'custom'
- 'box'
- 'composite'

### 3.6 Rozpoznanie transformacji geometrycznej

Polecenie `fitgeotrans` pozwala na estymację wartości parametrów transformacji (np. obrotu czy skalowania). W tym celu konieczne jest określenie par punktów na dwóch obrazach (przed i po operacji) odpowiadającym tej samej lokalizacji. Do tej operacji można użyć funkcji `cpselect`. W zależności od ilości wyznaczonych par punktów, możliwa jest estymacja wartości parametrów przekształceń:

- 'Nonreflective similarity' - dla dwóch i więcej par punktów
- 'Affine' - dla 3 i więcej par punktów
- 'projective' - dla 4 i więcej par punktów

### 3.7 Fraktale

Przekształcenia afiniczne można wykorzystać do generowania fraktali (J. Kudrewicz, *Fraktale i Chaos*, WNT, 1993). W tym celu wykonywane są losowo wybrane przekształcenia afiniczne na pojedynczej współrzędnej. Wartość elementu tablicy o obliczonych współrzędnych zostaje zwiększona o jeden. Przykładowy komplet tablic przekształceń afinicznych zamieszczono poniżej.

$$\begin{bmatrix} -0.67 & -0.02 & 0 \\ -0.18 & 0.81 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4 & 0.4 & 0 \\ -0.1 & 0.4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.4 & -0.4 & 0 \\ -0.1 & 0.4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.1 & 0 & 0 \\ 0.44 & 0.44 & -2 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

## Filtracja w domenie przestrzeni

### 1 Filtracja

Filtracja liniowa jest procesem splotu (konwolucji) obrazu z maską (filtrem). Dla dwuwymiarowej i dyskretniej funkcji filtracja dana jest wzorem:

$$L2(m, n) = (w \times L1)(m, n) = \sum_{p, q \in MF} L1(m - p, n - q)w(p, q), \quad (1)$$

gdzie  $w(p, q)$  - maska filtru MF

Do filtracji służy polecenie `imfilter(obraz, maska, opcje)`; Dostępne opcje są pogrupowane w 3 kategoriach:

- Opcje związane z brzegiem obrazu:
  - **X** - punkty leżące na zewnątrz obszaru, a pokryte maską, mają przypisaną wartość X. Domyślnie X=0;
  - 'symmetric' - potrzebne obszary są tworzone poprzez odbicie symetryczne
  - 'replicate' - potrzebne obszary są tworzone poprzez kopiowanie brzegowych wartości
  - 'circular' - potrzebne obszary są doklejane cyklicznie
- Opcje związane z rozmiarem wyjściowym:
  - 'same' - obraz wyjściowy ma taki sam rozmiar jak wejściowy
  - 'full' - obraz wyjściowy jest większy niż wejściowy
- algorytm filtracji
  - 'corr' - filtracja jest dokonywana z wykorzystaniem funkcji korelacji.
  - 'conv' - wielowymiarowa filtracja z wykorzystaniem splotu.

Poprzez maskę należy rozumieć macierz o wymiarach PxQ (najczęściej kwadratową o nieparzystym wymiarze), której element centralny wskazuje punkt obrazu cyfrowego podlegającemu filtracji.

#### 1.1 Rodzaje filtracji

Filtrację można podzielić na względem kilku kryteriów:

- wg pasma przepuszczania: np. dolno-, górno-, pasmoprzepustowe;
- wg liniowości: liniowe i nieliniowe.
- wg istnienia uprzywilejowanych kierunków działania: izotropowe, anizotropowe

### 2 Filtracja liniowa

#### 2.1 Filtracja dolnoprzepustowa

Przypisują one elementowi centralnemu średnią z obszaru pokrytego maską. Stosuje się do tego maski typu:  $MF = \text{ones}(3)/9$ ,  $MF = \text{ones}(5)/25$ . Działanie tego rodzaju filtru powoduje na ogół rozmycie krawędzi, generalizację obrazu oraz zawężenie zakresu intensywności. Cechą charakterystyczną tej filtracji jest, że suma elementów maski równa się 1. Najczęściej spotykane maski bazują na średniej ważonej, gdzie waga elementu centralnego jest większa od pozostałych, np.  $MF = [1 \ 1 \ 1; \ 1 \ 2 \ 1; \ 1 \ 1 \ 1]/10$ .

## 2.2 Filtracja górnoprzepustowa

Filtracja górnoprzepustowa służy do wyeksponowania detali i zmienności na obrazach, takich jak krawędzie, narożniki, pojedyncze punkty. Wykorzystuje się do tego pierwsze i drugie pochodne intensywności. Najczęściej używanymi operatorami są: gradienty Roberta, Sobela czy Prewitta oraz laplasjany.

### 2.2.1 Operator Prewitta

Operator ten jest oparty o pierwszą pochodną. Wykorzystuje on następujące maski (po lewej maska horyzontalna, po prawej wertykalna):

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

### 2.2.2 Operator Sobela

Operator ten jest oparty o uśrednioną pochodną dyskretną, gdzie wartość pochodnej liczonej przez środek ma dwukrotnie większą wagę niż pochodne liczone z elementów sąsiadujących. Wykorzystuje on następujące maski (po lewej maska horyzontalna, po prawej wertykalna):

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

### 2.2.3 Operator Roberta

Operatory Roberta to osiem masek, obracanych o 45 stopni względem siebie.

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}$$

### 2.2.4 Operator Nevatia-Babu

Operator Nevatia-Babu bazuje na obszarze 5x5 i wyznacza gradienty dla kierunków 0°, 30° i 60° (od lewej do prawej).

$$\begin{bmatrix} 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 \\ 0 & 0 & 0 & 0 & 0 \\ -100 & -100 & -100 & -100 & -100 \\ -100 & -100 & -100 & -100 & -100 \end{bmatrix} \quad \begin{bmatrix} 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 78 & -32 \\ 100 & 92 & 0 & -92 & -100 \\ 32 & -78 & -100 & -100 & -100 \\ -100 & -100 & -100 & -100 & -100 \end{bmatrix}$$

$$\begin{bmatrix} 100 & 100 & 100 & -32 & -100 \\ 100 & 100 & 92 & -78 & -100 \\ 100 & 100 & 0 & -100 & -100 \\ 100 & 78 & -92 & -100 & -100 \\ 100 & 32 & -100 & -100 & -100 \end{bmatrix}$$

### 2.2.5 Operator Kirscha

Operator Kirscha polega na przypisaniu elementowi centralnemu wartości maksymalnej z filtracji ośmioma maskami, obróconymi względem siebie o  $45^\circ$ . Pozwala to lepiej wyodrębnić maksymalną amplitudę z ośmiu kierunków filtracji. Pierwsze trzy maski dane są następującymi tablicami:

$$MF^0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad MF^{45} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad MF^{90} = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$$

### 2.2.6 Składanie filtracji

Ze względu na kierunkowość masek (ich anizotropowość), w celu wykrycia wszystkich krawędzi na obrazie konieczne jest składanie wyników filtracji, uzyskanych w wyniku działania ortogonalnych masek. Do najczęściej spotykanych metod składania można zaliczyć:

- wg Normy L2 - wartość filtracji uzyskujemy w wyniku złożenia przy użyciu normy Euklidesowej obrazów po filtracji:

$$Wynik = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (2)$$

- wg Normy Czebyszewa (maksimum) - wartość filtracji uzyskujemy w wyniku złożenia przy użyciu normy maksimum obrazów po filtracji:

$$Wynik = \max\left(\left|\frac{\partial I}{\partial x}\right|, \left|\frac{\partial I}{\partial y}\right|\right) \quad (3)$$

- wg Normy L1 (czasem 1/2 jest pomijana):

$$Wynik = \frac{1}{2} \left( \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right| \right) \quad (4)$$

### 2.2.7 Laplasjany

Lapsjan jest kombinacją drugich pochodnych cząstkowych funkcji  $I(m,n)$ : (wyostrzające, gradientowe):

$$L(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \quad (5)$$

Suma elementów maski laplasjanów może być równa 0 (wykrywanie krawędzi) lub 1 (wyostrzanie obrazu). Jest to filtracja izotropowa. Do najczęściej spotykanych masek laplasjanów zaliczamy:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0.25 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.25 \\ 0.5 & 0.5 & 0 & -0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & -0.75 & -1.5 & -0.75 & 0 & 0.5 \\ 0.5 & -0.5 & -1.5 & -2 & -1.5 & -0.5 & 0.5 \\ 0.5 & 0 & -0.75 & -1.5 & -0.75 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & -0.5 & 0 & 0.5 & 0.5 \\ 0.25 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.25 \end{bmatrix}$$

## 2.3 fspecial

Polecenie `fspecial(typ, parametry)` tworzy maski specjalne. Do parametrów funkcji `fspecial` należą:

- ('average', [rozmiar]) filtr uśredniający. Wartość domyślna to [3 3]. Maska filtru jest liczona według wzoru:

$$h = \text{ones}(N, M) / (N \cdot M) \quad (6)$$

- ('disk', promień) uśredniająca maska kołowa. Promień domyślnie = 5;
- ('gaussian', rozmiar, sigma) dolnoprzepustowy filtr gaussowski o symetrii kołowej. Wartości domyślne to [3 3] i  $\sigma = 0.05$ . Maska filtru **h** jest tworzona w następujący sposób:

$$h_g(n_1, n_2) = \exp \frac{-(n_1^2 + n_2^2)}{2\sigma^2} \quad (7)$$

$$h(n_1, n_2) = \frac{h_g(n_1, n_2)}{\sum_{n_1} \sum_{n_2} h_g(n_1, n_2)} \quad (8)$$

Zamiast tworzyć maskę i następnie filtrować, można skorzystać z funkcji `imgaussfilt`.

- ('laplacian', alpha) dwuwymiarowy laplasjan,  $\alpha \in (0, 1)$ , domyślnie 0.2. Maska jest tworzona w następujący sposób:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \approx \frac{1}{\alpha + 1} \begin{bmatrix} \alpha & 1 - \alpha & \alpha \\ 1 - \alpha & -4 & 1 - \alpha \\ \alpha & 1 - \alpha & \alpha \end{bmatrix}$$

- ('log', rozmiar, sigma) laplasjan filtru gaussowskiego. Rozmiar domyślny to [5 5], a  $\sigma = 0.5$ . Filtr **h** jest tworzony z wykorzystaniem wzoru 7:

$$h(n_1, n_2) = \frac{(n_1^2 + n_2^2 - 2\sigma^2)h_g(n_1, n_2)}{2\pi\sigma^6 \sum_{n_1} \sum_{n_2} h_g} \quad (9)$$

- ('motion', len, theta) aproksymacja przesuwania się kamery. Len określa długość przesunięcia (9) pod kątem  $\theta$  (domyślnie kąt=0)
- 'prewitt', 'sobel' - horyzontalna maska Prewitta, Sobela
- ('unsharp', alpha) - filtr rozmywający, działanie odwrotne do laplasjanu. Filtr jest tworzony na podstawie wzoru:

$$h = \frac{1}{\alpha + 1} \begin{bmatrix} -\alpha & \alpha - 1 & -\alpha \\ \alpha - 1 & \alpha + 5 & \alpha - 1 \\ -\alpha & \alpha - 1 & -\alpha \end{bmatrix}$$

## 2.4 Operatory krawędziowe kolorowe

Filtracja realizowana jest jako suma różnic trzech par punktów leżących symetrycznie względem punktu centralnego. Różnicę dwóch D punktów P1(R,G,B) i P2(R,G,B) możemy zdefiniować na 4 sposoby<sup>1</sup>:

<sup>1</sup>Opis kolorowych operatorów krawędziowych oraz działania filtru Canny pochodzi z: Kasprzak, W. (2009), *Rozpoznawanie obrazów i sygnałów mowy*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa

1. z wykorzystaniem normy Euklidesowej:

$$D(P1, P2) = \sqrt{(R1 - R2)^2 + (G1 - G2)^2 + (B1 - B2)^2} \quad (10)$$

2. z wykorzystaniem normy L1:

$$D(P1, P2) = |R1 - R2| + |G1 - G2| + |B1 - B2| \quad (11)$$

3. jako maksimum z wartości bezwzględnych poszczególnych składowych:

$$D(P1, P2) = \max(|R1 - R2|, |G1 - G2|, |B1 - B2|) \quad (12)$$

4. jako średnia ważona wartości bezwzględnych różnic poszczególnych składowych:

$$D(P1, P2) = \omega_1 \cdot |R1 - R2| + \omega_2 \cdot |G1 - G2| + \omega_3 \cdot |B1 - B2| \quad (13)$$

### 3 Filtracja nieliniowa

Istnieje również cała gama filtrów, których działanie zależy od np. uporządkowania obrazu. Są to np. filtry medianowe, maksymalizujące, minimalizujące czy entropii, które przypisują filtrowanemu punktowi odpowiednio medianę, maksimum, minimum i entropię z obszaru pokrytego maską. W pakiecie MatLAB 2014b zostały zdefiniowane następujące metody filtracji:

- `medfilt2(obraz, [rozmiar]);` - filtracja medianowa
- `ordfilt2(obraz, nr, zasięg);` - zwraca wartość znajdującą się na pozycji `nr` w uporządkowanym wektorze (1 - minimum, wielkość maski - maksimum).
- `rangefilt(obraz, zasięg)` - zwraca różnicę pomiędzy maksimum i minimum intensywności dla pikseli otoczenia zdefiniowanych w tablicy `zasięg`, np. `ones(3)`.
- `entropyfilt(obraz, zasięg);` - zwraca wartość entropii  $E$  w blokach o zadanym rozmiarze (domyślnie 9x9 pikseli), obliczoną wg wzoru:

$$E = - \sum_{n=0}^N \log p(n) \cdot p(n) \quad (14)$$

$N$  - ilość poziomów intensywności

$p(n)$  - prawdopodobieństwo wystąpienia piksela o intensywności  $n$

- `stdfilt(obraz, zasięg)` - dla każdego piksela zwraca wartość odchylenia standardowego z otoczenia.
- `modelfilt(obraz, [rozmiar])` - filtracja dominanty.

#### 3.1 Operacje blokowe

Inną możliwością tworzenia filtrów nieliniowych są tzw. operacje blokowe. Do implementacji tego typu filtrów służy polecenie `nlfilter(obraz, [rozmiar], funkcja)`, np.: `nlfilter(obraz, [3 3], 'min(x(:))')`.

Cechują się one znacznie dłuższym czasem działania niż gotowe funkcje, dlatego jeżeli istnieją gotowe funkcje filtrujące (jak np. `medfilt2`), należy z nich korzystać, a operacje blokowe stosować tylko w ostateczności.

### 3.2 Funkcja edge

Do wykrywania naroży może również służyć funkcja `edge(obraz, 'metoda', parametry)`. Funkcja ta na wejściu przyjmuje obraz monochromatyczny, a zwraca mapę logiczną, gdzie wartość logiczną "1" ma wykryty narożnik. Funkcja ta posiada 6 metod detekcji krawędzi (zarówno liniowych, jak i nieliniowych):

- `edge(obraz, 'Sobel', próg, 'kierunek', 'opcje')`. Wykorzystuje ona maskę Sobela. Dozwolone są następujące kierunki: 'horizontal', 'vertical', 'both'. Do opcji zaliczamy operację 'thinning' i 'nothinning'
- `edge(obraz, 'prewitt', próg, 'kierunek')`. Operator Prewitta, parametry analogicznie do Sobela.
- `edge(obraz, 'roberts', próg, 'kierunek', 'opcje')` - filtracja wykorzystująca operator Robertsa.
- `edge(obraz, 'log', próg, sigma)` - Laplasjan metody Gaussa. Parametr sigma określa standardowe odchylenie metody. Wartością domyślną jest 2. Wielkość maski filtru ( $n \times n$ ) jest liczona według wzoru:  $n = \text{ceil}(\sigma * 3) * 2 + 1$
- `edge(obraz, 'zerocross', próg, filtr_h)`
- `edge(obraz, 'canny', [ $\theta_1, \theta_2$ ], sigma)` - operator Canny.  $\theta_1$  określa dolny próg odcięcia,  $\theta_2$  określa górny. Jeżeli podany jest tylko jeden próg, to jest on traktowany jako górny, a dolny jest liczony według wzoru:  $\theta_1 = 0.4 \cdot \theta_2$ . Sigma jest to standardowe odchylenie dla filtru gaussowskiego. Domyślnie  $\sigma = 1$ . Filtr Canny działa w kilku krokach:

1. Wygładzanie obrazu za pomocą filtru Gaussa o wariancji  $\sigma^2$
2. Wyznaczenie obrazu krawędziowego prostym operatorem krawędziowym w obszarze 2x2 lub 3x3
3. Cieniowanie krawędzi metodą tłumienia niemaksymalnego - zerowanie elementów krawędziowych niebędących maksimumami lokalnymi
4. Cieniowanie krawędzi metodą tłumienia z progami histerezy  $\theta_1, \theta_2$

Dla wszystkich tych metod, poleceniem `[BW, próg]=edge(...)` można uzyskać informacje o użytym progu detekcji krawędzi. Dla metod gradientowych (Sobel, Prewitt, Roberts) automatyczny próg binaryzacji jest liczony na podstawie magnitudy policzonego gradientu.

### 3.3 VMF - Vector median flow

Jest to filtracja stosowana wyłącznie dla obrazów wielokanałowych (np. RGB, CMYK, satelitarne). Polega ona na zastąpieniu wartości piksela centralnego przez wartość koloru leżącego najbliżej w przestrzeni kolorów. Jeżeli w masce istnieje piksel o identycznej wartości koloru jak piksel centralny, to wartość piksela nie ulega zmianie. Odległość dwóch wartości kolorów w danej przestrzeni najczęściej liczona jest z użyciem normy L1 lub L2.

## 4 Filtracja adaptacyjna

Osobną kategorią filtrów nieliniowych są filtry adaptacyjne. Są to filtry, których charakterystyka zmienia się w zależności od analizowanego obszaru.

Dolnoprzepustowa **filtracja Wienera** działa dwuetapowo:

1. Dla każdego punktu i jego otoczenia obliczamy wartość parametru, który kwalifikuje dany punkt jako należący lub nie do krawędzi.
2. jeżeli dany punkt został zakwalifikowany jako nie należący do krawędzi, zostaje on poddany silnemu uśrednieniu. W przeciwnym wypadku jego wartość pozostaje bez zmian lub poddany zostaje uśrednieniu o niewielkiej mocy.

Realizujemy ją poleceniem: `wiener2(obraz, [rozmiar])`.

Innym rodzajem jest **adaptacyjny filtr medianowy**. Działa on w następujący sposób:

1. Dla każdego piksela obliczamy średnią i odchylenie standardowe w pewnej masce.
2. Jeżeli wartość intensywności piksela centralnego mieści się w zakresie  $\text{średnia} \pm (\text{współczynnik} \cdot \text{odchylenie})$ , to wartość piksela nie ulega zmianie.
3. Jeżeli wartość piksela jest poza przedziałem, to zastępowana jest on wartością medianą z wartości pikseli otoczenia mieszczących się we wspomnianym przedziale.

Ilość zmian zależy od wielkości maski oraz od wartości współczynnika. Jego wartość w większości przypadków nie przekracza wartości 2 (zwykle wynosi 1).

## 5 Dekonwolucja

Operacją odwrotną do splotu (czyli m.in. filtracji) jest dekonwolucja. Istnieje kilka technik dekonwolucji zaimplementowanych w MatLABie:

- Dekonwolucja ślepa: `deconvblind`
- Dekonwolucja przy użyciu algorytmu Lucy-Richardsona: `deconvlucy`
- Dekonwolucja przy użyciu algorytmu Wienera: `deconvwnr`
- Dekonwolucja "unormowana": `deconvreg`

## 6 Szum

Do "zaszumiania" obrazów służy polecenie `imnoise(obraz, typ, parametry)`. Podstawowymi rodzajami szumu są:

- 'gaussian' - szum o rozkładzie gaussa z średnią  $m$  i wariancją  $\sigma^2$ . Domyślnie  $m=0$  i  $\sigma^2=0.01$ ;
- 'localvar' - szum gaussowski o średniej  $m=0$  i wariancji zależnej od otoczenia punktu. Aby użyć tej funkcji należy jako parametr dodać tablicę rozkładu wariancji o rozmiarze równym zaszumionemu obrazowi.
- 'poisson' - szum o rozkładzie Poissona
- 'salt & pepper' - (szum impulsowy) zamienia wartość piksela na minimum lub maksimum z dostępnej skali z gęstością zmiany  $d$ . Domyślnie  $d=0.05$ ;
- 'speckle' - multiplikatywny szum zmieniający wartość piksela zgodnie z równaniem:

$$L(m, n) = L(m, n) + n \cdot L(m, n) \quad (15)$$

gdzie  $n$  - liczba z rozkładu normalnego o średniej  $m=0$  i wariancji  $\sigma^2$ . Domyślnie  $\sigma^2=0.04$ ;



## Przekształcenia morfologiczne

### 1 Element strukturalny

Przekształcenia morfologiczne są to operacje związane z zastosowaniem tzw. Elementu strukturalnego (SE). Jest to pewien wycinek obrazu z wyróżnionym jednym punktem (elementem centralnym). Element strukturalny jest przemieszczany po całym obrazie i dla każdego punktu wykonywana jest analiza koincydencji punktów obrazu i elementu strukturalnego.

Do tworzenia elementu strukturalnego służy polecenie `strel( 'kształt', parametry );`. Możliwe są następujące kształty:

- ('arbitrary', NHOOD); - dowolny element definiowany poprzez macierz NHOOD;
- ('pair', Offset) - zawiera dwa elementy oddalone o offset
- ('diamond', R) - kwadrat obrócony o  $45^\circ$ . R - połowa przekątnej
- ('periodicline', P, V) -  $(2*P+1)$  punktów wzdłuż linii oddalonych o wektor V między sobą.
- ('disk', R) - koło o promieniu R
- ('rectangle', [M N]) - prostokąt wypełniony jedynkami
- ('line', Len, Deg) - linia o długości Len i nachyleniu Deg
- ('square', w) - kwadrat o boku w
- ('octagon', R) - ośmiokąt o promieniu R

### 2 Erozja

Aby zdefiniować erozję należy założyć, że istnieje nieregularny obszar X i koło B o promieniu r. Zerodowana figura X elementem B jest to zbiór wszystkich środków kół o promieniu r, które w całości zawarte są wewnątrz obszaru X. W implementacji komputerowej polega to na usunięciu wszystkich elementów o wartości 1, które posiadają choć jednego sąsiada o wartości 0. Można to interpretować jako filtr minimalny. Do erozji służy polecenie `imerode(obraz, SE)`. Jako elementów strukturalnych używa najczęściej następujących masek:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & X & X \\ X & 1 & X \\ X & X & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

gdzie X - dowolna wartość obrazu  $\{0,1\}$

W MATLABie poszczególne elementy koduje się następująco:

1 to 1; 0 to -1; X to 0.

#### 2.1 Cechy erozji

Do najważniejszych cech erozji należą:

- w większości przypadków jest addytywna, tzn. erozję o założonej wielkości można interpretować jako złożenie odpowiedniej ilości erozji o mniejszej wielkości. Twierdzenie to jest nieprawdziwe przy bardzo małych elementach strukturalnych, których kształt istotnie odbiega od teoretycznego elementu strukturalnego;

- erozja złożonym SE jest równoważna złożeniu erozji poszczególnymi elementami tego SE;
- położenie punktu centralnego SE nie ma większego znaczenia. Zmiana położenia punktu centralnego o dany wektor powoduje przesunięcie obrazu wynikowego o ten sam wektor;
- erozja ma zdolność do eliminacji drobnych szczegółów i wygładzania brzegu figury;
- erozja elementem o podłużnym kształcie pozwala uwypuklić linijne fragmenty zorientowane w tym samym kierunku, co SE;
- erozja dokonuje generalizacji obrazu.

### 3 Dylatacja

Jest to przekształcenie odwrotne do erozji. Zachowując założenia z erozji, figura po dylatacji jest zbiorem środków wszystkich kół B, dla których choć jeden ich punkt pokrywa się z jakimkolwiek punktem figury X. Analogicznie do erozji, można to przekształcenie traktować jako filtr maksymalizujący. Dylatację można dodatkowo zdefiniować jako negatyw erozji negatywu obrazu. Do dylatacji służy polecenie `imdilate(obraz, SE)`; Typowy element strukturalny dylatacji wygląda następująco:

$$\begin{bmatrix} X & X & X \\ X & 0 & X \\ X & X & X \end{bmatrix}$$

Przy czym przynajmniej jeden element X musi mieć wartość 1.

#### 3.1 Cechy Dylatacji

Do najważniejszych cech dylatacji należą:

- zamykanie małych otworów i wąskich zatok
- zdolność do łączenia obiektów blisko połączonych
- podobnie jak w przypadku erozji addytywność i możliwość zamiany dylatacji złożonym SE na dylatację elementami składowymi.

### 4 Otwarcie i Zamknięcie

Są to operacje będące złożeniem erozji i dylatacji. Otwarcie polega na następujących po sobie erozji i dylatacji tym samym elementem, a zamknięcie następującej po sobie dylatacji i erozji:

$$\begin{aligned} \text{otwarcie} &= \text{dylatacja}(\text{erozja}, \text{SE}), \text{SE}) \\ \text{zamknięcie} &= \text{erozja}(\text{dylatacja}, \text{SE}), \text{SE}) \end{aligned}$$

Do wykonania tych operacji służą następujące polecenia:  
`imopen(obraz, SE);` i `imclose(obraz, SE);`.

Otwarcie powoduje usunięcie drobnych obiektów takich jak np. półwyspy, wypustki, nie zmieniając zasadniczo wielkości figury. Może rozłączyć niektóre obiekty połączone pierwotnie przewężeniami. Zamknięcie wypełnia wąskie zatoki oraz drobne otwory wewnątrz figury. Może też połączyć leżące blisko siebie obiekty.

## 4.1 Cechy Otwarcia i Zamknięcia

Najważniejszymi cechami Otwarcia  $O(F)$  i Zamknięcia  $C(F)$  są:

- Przekształcenia te są niezmiennie wobec siebie:  $O(O(F))=O(F)$  i  $C(C(F))=C(F)$
- Otwarcie usuwa drobne obiekty i szczegóły, może też rozłączyć niektóre obiekty z przewężeniami
- Zamknięcie wypełnia wąskie zatoki i wcięcia oraz drobne otwory wewnątrz obiektu. Może też połączyć leżące blisko siebie elementy.
- Zależność pomiędzy polem powierzchni figury  $F$ , a wynikiem Eroзии, Otwarcia, Zamknięcia i Dylatacji tej figury jest następująca:

$$E(F) \leq O(F) \leq C(F) \leq D(F) \quad (1)$$

## 5 Gradient morfologiczny

Gradient morfologiczny jest definiowany na trzy sposoby:

- jako różnica między obrazem wejściowym, a wynikiem jego eroзии;
- jako różnica między wynikiem dylatacji, a obrazem wejściowym;
- jako różnica między wynikiem dylatacji obrazu wejściowego, a jego erozią. W tym wariancie wynik często bywa dzielony przez 2.

## 6 Rekonstrukcja

Rekonstrukcja obrazu przy użyciu markera. Obraz i marker mogą być obrazami logicznymi lub monochromatycznymi i muszą mieć ten sam rozmiar. Rekonstrukcja polega na cyklicznym dokonywaniu dylatacji obrazu i wyznaczaniu części wspólnej obrazu uzyskanego po dylatacji i obrazu wejściowego dla całego przekształcenia. Cykl ten powtarza się aż do uzyskania zbieżności, to znaczy braku zmian pomiędzy sąsiednimi iteracjami.

W MatLABie używane jest do tego polecenie:

`imreconstruct(marker, obraz, sąsiedztwo).`

## 7 Operacja Hit or Miss

Operacja Hit or miss (trafi, nie trafi) jest definiowane jako:

"Do każdego punktu analizowanego obrazu przykładany jest punkt centralny danego elementu strukturalnego. Jeżeli lokalne otoczenie analizowanego punktu zgodne jest z elementem strukturalnym - odpowiedni punkt obrazu wynikowego uzyskuje wartość 1. W przeciwnym przypadku wartość 0." (Tadeusiewicz & Kohoroda). Stosuje się różne elementy strukturalne, mające różne zastosowanie. I tak:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ X & X & X \end{bmatrix} \quad \begin{bmatrix} X & 1 & X \\ X & 1 & X \\ 1 & X & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & X & X \\ X & 1 & X \\ 1 & X & 1 \end{bmatrix}$$

Pierwszy element służy do detekcji pojedynczych izolowanych punktów, drugi do detekcji punktów końcowych, a pozostałe dwa do detekcji punktów węzłowych.

Do użycia tego przekształcenia służy polecenie `bwhitmiss(obraz, SE1, SE2)` lub `W=bwhitmiss(A, SE);`

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad SE = \begin{bmatrix} 1 & 0 & -1 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \quad W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

## 8 Wypukłe otoczenie

Wypukłe otoczenie jest to najmniejsza figura wypukła zawierająca daną figurę. Realizuje się to przez pogrubienie maską, obracaną co  $45^\circ$ :

$$\begin{bmatrix} 1 & 1 & X \\ 1 & 0 & X \\ 1 & X & 0 \end{bmatrix}$$

## Przekształcenia morfologiczne II i operacje na obrazach logicznych

### 1 Przekształcenia morfologiczne

#### 1.1 Ścienianie i Pogrubianie

Pogrubianie i ścienianie: Operacje te polegają na nałożeniu lub ściągnięciu wierzchniej warstwy obiektu. W procesie ścieniania wartość punktu nie zmienia się, gdy SE nie pokrywa się z jego sąsiedztwem lub zmienia jego wartość na zero, gdy SE pasuje do sąsiedztwa rozpatrywanego punktu. W procesie pogrubiania relacje są odwrotne. Do wykonywania tych operacji służy polecenie `bwmorph` z parametrami `'thicken'` i `'thin'`: `bwmorph(obraz, 'thin')`;

Elementem wykorzystywanym do ścieniania jest maska, obracana co  $90^\circ$ :

$$\begin{bmatrix} X & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

#### 1.2 Szkieletyzacja

Szkielet figury to zbiór wszystkich wszystkich środków okręgów, mieszczących się w całości wewnątrz figury i mających co najmniej dwa punkty wspólne z brzegiem figury. Szkieletyzację wykonuje się przy użyciu następujących elementów:

$$\begin{bmatrix} 1 & 1 & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & X & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & X \\ 1 & 1 & 0 \\ 1 & 1 & X \end{bmatrix}$$

Do szkieletyzacji służy polecenie `bwmorph(obraz, 'skel')`;

#### 1.3 Dylatacja bez stykania obszarów (SKIZ)

SKIZ (ang. skeleton by influence zone) jest pomocne przy rozdzielaniu elementów częściowo się stykających. Jako pierwszy krok wykonuje się erozję, a następnie pogrubienie następującym SE:

$$\begin{bmatrix} X & X & X \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Gdy przekształcenie to jest stosowane cyklicznie, aż do braku zmian, uzyskuje się szkielet wpływów (SKIZ). Strefa wpływów danego punktu definiowana jest jako zbiór wszystkich punktów obrazu, dla których odległość do danego punktu jest mniejsza niż do pozostałych. Przekształcenie to czasami powoduje pewne artefakty (wąskie i głębokie zatoki), które eliminuje się w 3 krokach: negatyw  $\rightarrow$  obcięcie gałęzi  $\rightarrow$  negatyw.

## 1.4 Obcinanie gałęzi

Do obcinania gałęzi, powstałych np. w wyniku szkieletyzacji lub wycieniania służą następujące elementy strukturalne:

$$\begin{bmatrix} X & 0 & 0 \\ 1 & 1 & 0 \\ X & 1 & X \end{bmatrix} \quad \begin{bmatrix} 0 & X & X \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

## 1.5 Pakiet bwmorph

`bwmorph(obraz, 'metoda', parametr)`; Funkcja ta aplikuje operacje morfologiczne na obrazie logicznym. Do metod zaliczamy:

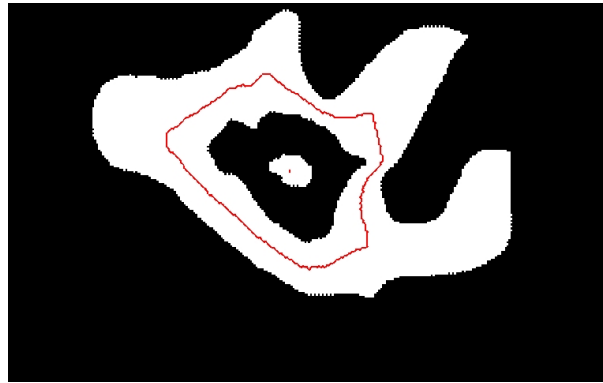
- `'bothat'` - operacja Bottom hat, która polega na odjęciu od zamknięcia obrazu samego obrazu. Istnieje jej uogólnienie na obrazy monochromatyczne `imbothat`.
- `'bridge'` - łączy blisko siebie leżące piksele.
- `'clean'` - zeruje pojedynczego piksela, otoczonego samymi zerami.
- `'close'` i `'open'` - zamknięcie i otwarcie elementem `ones(3)`.
- `'diag'` - zamienia połączenia diagonalne (8-sąsiedztwo) na 4-sąsiedztwo poprzez zmianę wartości piksela leżącego pod przekątną na 1.
- `'dilate'` i `'erode'` - dylatacja i erozja elementem `ones(3)`
- `'fill'` - zmienia wartość zerowego piksela otoczonego samymi jedynkami na jeden.
- `'hbreak'` - rozdziela H-połączone piksele (w pionie i poziomie)

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

- `'majority'` - jeżeli suma maski 3x3 jest większa lub równa 5, to element centralny przyjmuje wartość 1. Jeżeli nie, to element centralny ma wartość 0.
- `'remove'` - zmienia wartość środkowego piksela na 0, jeżeli wszyscy jego 4-sąsiedzi mają wartość 1.
- `('shrink',n)` - kurczy obiekty do punktów. W przypadku obiektu z "dziurami", i użycia tej funkcji z `n=inf`, powstaje obwódka w pomiędzy granicą zewnętrzną i wewnętrzną obiektu. Obiekty jednorodne są zmniejszane do punktu. Parametr `n` oznacza ilość iteracji, przy `n=inf` operacja jest wykonywana do momentu, kiedy pomiędzy wynikami dwóch sąsiednich iteracji nie będzie różnicy. Efekt działania tej funkcji z parametrem `n=inf` (kolor czerwony) naniesiony na obraz wejściowy przedstawia fig.1. Widoczna jest obwódka wokół dziury oraz pojedynczy punkt w wewnętrznym obiekcie.
- `('skel',n)` - szkieletyzacja. Działanie parametru `n` jest analogiczne do funkcji `'shrink'`.
- `'spur'` - usuwa gałęzie, połączone diagonalnie.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- `('thicken', n)` i `('thin', n)` - pogrubianie i wycienianie obiektu



**Rysunek 1:** Efekt działania funkcji `bwmorph('shrink',inf)` zaznaczony kolorem czerwonym

- **'tophat'** - operacja Top hat. Funkcja zwraca różnice między obrazem, a jego otwarciem. Istnieje jej uogólnienie na obrazy monochromatyczne `imtophat`.

## 1.6 Operacje geodezyjne

Operacje geodezyjne wymagają użycia dwóch obrazów (obraz wejściowy oraz maska / marker). W wyniku tych operacji obraz wejściowy jest poddawany operacji morfologicznej, a następnie porównywany z maską. Wyróżniamy następujące operacje geodezyjne:

- **dylatacja geodezyjna (o rozmiarze 1)** - jest to wartość minimalna z obrazu wejściowego poddanego dylatacji jednostkowym elementem strukturalnym oraz maski. W przypadku obrazów logicznych jest to iloczyn logiczny maski oraz obrazu po dylatacji. Zakłada się, że  $D(\text{obraz}) \leq \text{maska}$
- **erozja geodezyjna (o rozmiarze 1)** - jest to wartość maksymalna z obrazu wejściowego poddanego erozji jednostkowym elementem strukturalnym oraz maski. W przypadku obrazów logicznych jest to suma logiczna maski oraz obrazu po erozji. Zakłada się, że  $E(\text{obraz}) \geq \text{maska}$
- **rekonstrukcja przez dylatację** - wykonywanie cyklicznie dylatacji geodezyjnych, aż do braku różnic pomiędzy kolejnymi dylatacjami. Jest to najczęściej stosowana rekonstrukcja morfologiczna.
- **rekonstrukcja przez erozję** - wykonywanie cyklicznie erozji geodezyjnych, aż do uzyskania braku różnicy pomiędzy kolejnymi iteracjami.
- **otwarcie przez rekonstrukcję** - Polega na wykonaniu erozji zadany elementem strukturalnym, a następnie rekonstrukcji przez dylatację.
- **zamknięcie przez rekonstrukcję** - Polega na wykonaniu dylatacji zadany elementem strukturalnym, a następnie rekonstrukcji przez erozję.

## 2 Operacje na obrazach logicznych

Do innych operacji (niż morfologiczne) na obrazach logicznych należą następujące przekształcenia i operacje:

## 2.1 Estymacja pola powierzchni

Polecenie `bwarea(obraz)`; podaje estymowane pole powierzchni. Wartość uzyskana tą funkcją może być inna od wartości rzeczywistej (uzyskanej funkcją `sum(obraz(:))`). Zliczania dokonuje się w otoczeniu 2x2 i na podstawie następujących warunków:

- 0 - jeżeli w otoczeniu wszystkie piksele mają wartość 0.
- 0.25 - jeżeli tylko jeden piksel ma wartość 1.
- 0.5 - jeżeli dwa piksele mają wartość 1 i stykają się bokami.
- 0.75 - jeżeli dwa piksele mają wartość 1 i stykają się rogami.
- 0.875 - jeżeli 3 piksele mają wartość 1.
- 1 - jeżeli wszystkie 4 piksele mają wartość 1.

## 2.2 Wyznaczanie obwodu

Polecenie `bwperim(obraz, sąsiedztwo)`; wyznacza linie obwodu. Sąsiedztwo dla obrazów 2D może przyjmować wartość 4 lub 8. Rozbudowaniem tej funkcji jest polecenie

`D=bwboundaries(obraz, sąsiedztwo, opcje)`.

Jako rezultat działania funkcji otrzymujemy wektor o rozmiarze równym ilości izolowanych obiektów, składający się z tablic współrzędnych każdej granicy. Jako opcje mamy dwie możliwości: `'noholes'` i `'holes'`. Na fig.2 pokazano efekt działania tych parametrów. Kolorem czerwonym zaznaczono granice obliczone z parametrem `'noholes'`. Dzięki zastosowaniu opcji `'holes'` zyskujemy, oprócz dwóch czerwonych, dodatkową granicę zaznaczoną kolorem niebieskim.



**Rysunek 2:** Efekt działania funkcji `bwboundaries` z parametrami `'noholes'` - kolor czerwony, i `'holes'` - czerwony + niebieski

## 2.3 Obliczanie odległości

Polecenie `bwdist(obraz, 'metryka')`; podaje odległość od najbliższego piksela o wartości 1, przy wykorzystaniu jednej z następujących metryk:

- `'euclidean'`:  $d = \sqrt{(m_1 - m_2)^2 + (n_1 - n_2)^2}$
- `'quasi-euclidean'`:

$$d = \begin{cases} |m_1 - m_2| + (\sqrt{2} - 1)|n_1 - n_2| & \text{dla } |m_1 - m_2| > |n_1 - n_2| \\ (\sqrt{2} - 1)|m_1 - m_2| + |n_1 - n_2| & \text{dla } |m_1 - m_2| \leq |n_1 - n_2| \end{cases}$$



- 'chessboard':  $d = \max(|m_1 - m_2|, |n_1 - n_2|)$
- 'cityblock':  $d = |m_1 - m_2| + |n_1 - n_2|$

## 2.4 Liczba Eulera

`bweuler(obraz, sąsiedztwo)`; - podaje wartość liczby Eulera. Jest to ilość obiektów występujących na obrazie binarnym, pomniejszona o ilość dziur w tych obiektach. Sąsiedztwo przyjmuje wartość 4 lub 8.

## 2.5 Wypełnianie dziur w obiektach

Do wypełniania dziur w obiektach służy polecenie:

```
imfill(obraz, sąsiedztwo, lokalizacja, 'holes');
```

We wcześniejszych wersjach biblioteki jest to funkcja `bwfill`. W przypadku obrazów logicznych wypełnia wskazane myszką lub poprzez podanie współrzędnych punktu obszaru złożonego z zer wewnątrz figury. Dodanie parametru 'holes' wypełnia wszystkie dziury na obrazie. W przypadku obrazu monochromatycznego poprzez dziurę rozumie się obszary zbudowane z pikseli o mniejszej intensywności niż otoczenie.

## 2.6 Etykietowanie i segmentacja

Polecenie `bwlabel(obraz, sąsiedztwo)`; służy do etykietowania, tzn. do przypisywania jednakowej wartości pikselom wewnątrz obszarów jednorodnych i rozłącznych z innymi obiektami. Sąsiedztwo przyjmuje wartość 4 lub 8. Każdy jednorodny obiekt ma unikalny numer, będący kolejnymi dodatnimi liczbami całkowitymi.

Do prezentacji wyników etykietowania służy polecenie `label2rgb (wynik, 'paleta', [kolor zerowy], 'kolejność')`. Do etykietowania obrazów wielowymiarowych służy polecenie `bwlabeln (obraz, sąsiedztwo)`.

Z pojęciem etykietowania związana jest segmentacja czyli proces podziału obrazu na fragmenty odpowiadające widocznym na obrazie obiektom. Wyróżniamy dwie grupy metod segmentacji:

- poprzez podział obszaru: polega na stopniowym, iteracyjnym, podziale dużych obszarów na mniejsze, których piksele mają wartość różniącą się od sąsiedztwa.
- poprzez rozrost obszaru: piksele sąsiedztwa obszaru są sprawdzane czy spełniają warunki podobieństwa i w przypadku spełnienia, są dołączane do obszaru.

## 2.7 Wskazywanie obiektów

Polecenie `bwselect(obraz, x, y, sąsiedztwo)`; służy do wskazywania obszarów jednorodnych poprzez wskazanie dowolnego punktu wewnętrznego. Wektory x,y wskazują położenie pikseli wewnętrznych. W przypadku ich braku, wskazywanie odbywa się przy użyciu myszki. Sąsiedztwo=4,8 i oznacza typ sąsiedztwa. Wynikiem działania tego polecenia jest mapa logiczna, gdzie wartość jeden mają tylko obiekty wskazane poprzez punkty wewnętrzne.

## 2.8 Dział wodny

Dział wodny jest to linia rozdzielająca dwa obszary przyciągania (zlewnie). Służy do tego polecenie `watershed(obraz, sąsiedztwo)`; . Funkcja działu wodnego może służyć do rozdzielania blisko siebie leżących obiektów.

## Wprowadzenie do analizy obrazu<sup>1</sup>

Analiza obrazu polega na wydobyciu cech opisujących obraz. W jej skład wchodzi następujące procesy: segmentacja, lokalizacja obiektów oraz wyznaczenie ich cech.

### 1 Schemat analizy

Proces analizy obrazu jest ostatnim etapem całego procesu wydobywania informacji. Schemat na ogół wygląda następująco:

1. Akwizycja obrazu
2. Wstępne przetwarzanie (w większości przypadków jest to ogół operacji na obrazach kolorowych i monochromatycznych). Ma na celu poprawę jakości obrazu, uwypuklenie cech obiektów będących celem późniejszej analizy.
3. Segmentacja, binaryzacja i przetwarzanie obrazu binarnego (filtracja, rozłączenie elementów stykających się, usunięcie obiektów przeciętych brzegiem)
4. Analiza obrazu - segmentacja oraz następnie wyznaczenie parametrów obiektów (np. rozkładów powierzchni ziaren, średnic Fereta)
5. Wizualizacja wyników.

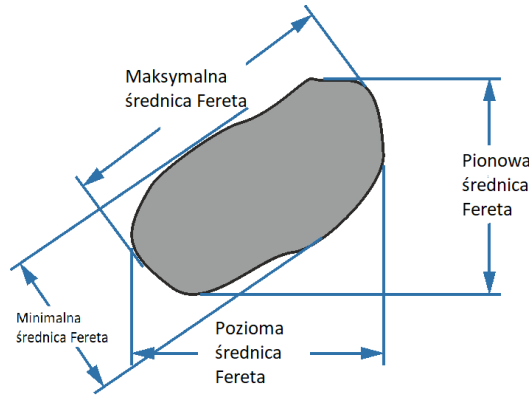
### 2 Parametry obiektów

Do najważniejszych parametrów obiektów zaliczamy :

- Liczebność obiektów (z uwzględnieniem efektu przecięcia brzegu). Jest to realizowane poprzez proces etykietowania.
- Liczba Eulera: liczba obiektów pomniejszona o ilość dziur w tych obiektach.
- Pola powierzchni, sumaryczne i rozkłady.
- Długości krawędzi (np. poprzez zliczanie punktów brzegowych, przybliżanie figury wielokątem, średniej z wewnętrznej i zewnętrznej krawędzi, formuły Croftona, ...)
- Długości rzutów: Rzutem figury  $D(\alpha)$  w kierunku wektora rzutowania  $\alpha$  nazywamy największą odległość pomiędzy wszystkimi prostymi, równoległymi do wektora  $\alpha$ , mającymi część wspólną z figurą  $D$ .
- Średnica zastępcza: średnica koła o polu powierzchni równym polu figury.
- Średnice Fereta: Długość najdłuższego oraz najkrótszego rzutu figury. Czasami wprowadza się też poziomą i pionową średnicę Fereta rozumianą jako długość rzutu poziomego i pionowego (Rys. 1).
- Wymiary fraktalne (np. kostkowy, korelacyjny, ...) - pozwalają na ocenę "poszarpania" obiektu.

---

<sup>1</sup>W większości na podstawie Tadeusiewicz, R., Korohoda, P. 1997 Komputerowa analiza i przetwarzanie obrazów



**Rysunek 1:** Średnice Fereta

- Momenty bezwładności n-tego rzędu dla pola powierzchni  $A(X)$  figury  $X$  :

$$M_{1X} = \frac{1}{A(X)} \sum_X x_i \quad (1)$$

$$M_{1Y} = \frac{1}{A(X)} \sum_X y_i \quad (2)$$

$$M_{2X} = \frac{1}{A(X)} \sum_X (x_i - M_{1X})^2 \quad (3)$$

$$M_{2Y} = \frac{1}{A(X)} \sum_X (y_i - M_{1Y})^2 \quad (4)$$

$$M_{2XY} = \frac{1}{A(X)} \sum_X (x_i - M_{1X})^2 \cdot (y_i - M_{1Y})^2 \quad (5)$$

- Współczynniki kształtu:

- Bezwymiarowy współczynnik kształtu ( $L$  - obwód,  $S$  - powierzchnia obiektu):

$$R_S = \frac{4\pi \cdot S}{L^2} \quad (6)$$

- Współczynnik Fereta  $R_F$  ( $F_{max}, F_{min}$  - maksymalna i minimalna średnica Fereta )

$$R_F = \frac{F_{min}}{F_{max}} \quad (7)$$

- Współczynnik cyrkularności:

$$R_{C1} = 2\sqrt{\frac{S}{\pi}}, \quad R_{C2} = \frac{L}{\pi} \quad (8)$$

- Współczynnik Blaira-Blissa

$$R_B = \frac{S}{\sqrt{2\pi \sum_i r_i^2}} \quad (9)$$

gdzie:

$S$  - pole powierzchni

$r_i$  - odległość  $i$ -tego piksela od środka ciężkości

## Projekt I

### 1 Opis Projektu

W języku C / C++ / C# (inne języki programowania - uwaga poniżej) należy napisać program, który:

- Wczytuje zadany obraz, umożliwia przetwarzania wg tabelki, a następnie wyniki przekształceń zapisuje do plików.
- Zabronione jest korzystanie z gotowych bibliotek funkcji obrazów w sposób inny niż do odczytu i zapisu z pliku, alokacji miejsca, pobrania/zapisania wartości piksela oraz wyświetlenia (elementy GUI). W szczególności zabronione jest korzystanie gotowych funkcji stricte obrazowych (np. przekształcenia punktowe, morfologiczne, etc.).
- Dozwolone jest korzystanie z gotowej implementacji FFT w wersji 1D/2D.
- Każde przekształcenie ma być przetestowane na dwóch różnych obrazach (za wyjątkiem logicznych i oznaczonych jako "tylko monochromatyczne"), z czego jeden ma być monochromatyczny, a drugi wg. tabeli.
- Wszystkie operacje mają być zaimplementowane w ramach jednego programu, a nie jako cztery niezależne aplikacje. Wygląd aplikacji, wydajność kodu, etc. **nie wpływają** na ocenę z projektu. Program ma tylko i aż działać poprawnie. Mają działać wszystkie polecenia - nie ma ocen częściowych za działanie np. 3 z 4 procedur.
- Zaliczenie projektu jest warunkiem koniecznym, nie wystarczającym, do uzyskania pozytywnej oceny z ćwiczeń.
- Słowo "**zadany**" oznacza, że wartość danego parametru jest podawana w czasie wykonywania programu, a nie wpisana na sztywno w kod programu.
- W raporcie mają być zamieszczone wyniki przekształceń oraz opis ich działania. Raport ma być wysłany w formacie **PDF** na adres: dwornik@agh.edu.pl

W zależności od terminu nadesłania gotowych projektów (PDF + spakowane kody), projekt będzie oceniony na maksymalnie:

- do 07.05.2023 - 8 pkt;
- 08.05-18.06.2023 - 5 pkt;
- 18.06-15.09.2023 - 1 pkt;
- od 16.09.2023 - brak zaliczenia.

W przypadku pisania w innych językach programowania niż C / C++ / C#, należy dodatkowo pokazać (np. poprzez spotkanie via MS Teams, konsultacje lub bezpośrednio po zajęciach), że dany kod kompiluje się i działa poprawnie. Niestety nie jestem w stanie mieć każdego kompilatora i każdej biblioteki, a nie zamierzam blokować Państwa kreatywności. JAVA / Python jest akceptowalna (obowiązuje wymóg pokazania).

## 2 Tabela przekształceń

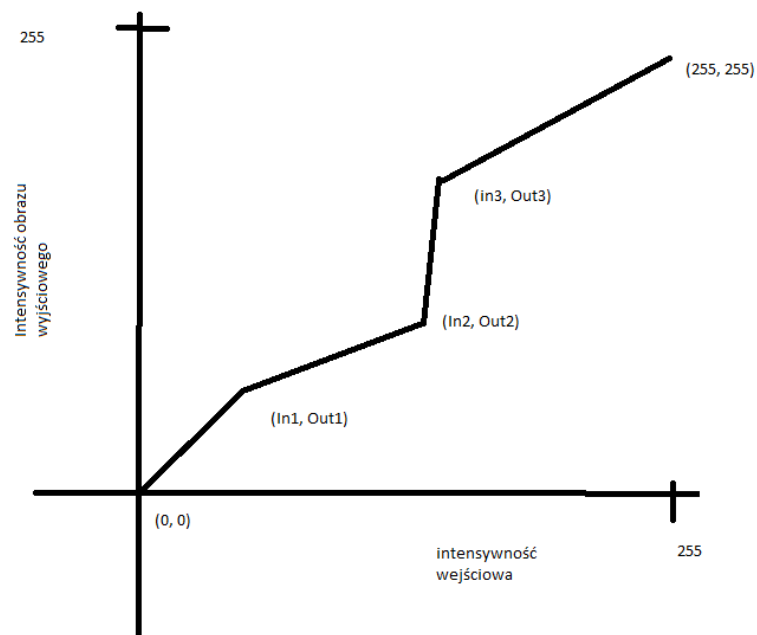
Ostatnia cyfra indeksu	Punktowe, Geometryczne	Filtracja przestrzenna	Przedostat nia cyfra indeksu	Morfologiczne	Logiczne
	RGB+mono	RGB + mono		mono + logiczne	tylko log.
1,2	Regionprops (tylko monochrom): Centroid, ekwiwalent średnicy (ilość pikseli = pole koła). Zapis wyników do pliku txt.	Ordfilt2 dla zadanego rozmiaru mask i zadanego numeru porządkowego. Dla RGB każda warstwa osobno	1,2	Zamknięcie elementem kołowym o zadany promieniu	Mapa odległości geodezyjnej od zadanego punktu wewnątrz całego obiektu
3,4	Normalizacja obrazu wg łamanej opisanej przez min. 3 zadane pary punktów (intensywność wejścia, intensywność wyjścia) - rys.2	Filtracja entropii w zadany oknie. Dla RGB entropia liczona dla potrójnego zbioru danych(*). Znormalizować	3,4	Otwarcie elementem kołowym o zadany promieniu	Wypełnianie dziur w obiektach (przez rekonstrukcję)
5,6	Generowanie fraktali metodą z rozd. 2.7 (punktowe.pdf) Tylko mono.	Filtracja Kirscha. Brzeg odbicie symetryczne. Dla RGB każda warstwa osobno	5,6	Gradient morfologiczny elementem "+" 3x3 każdą z 3 metod	Usunięcie elementów przeciętych brzegiem
7,8	Wyrównanie histogramu do rozkładu Gaussa o zadany odchyleniu (rys. 3)	Filtracja entropii w zadany oknie. Dla RGB entropia liczona dla potrójnego zbioru danych(*). Znormalizować	7,8	Otwarcie elementem linijnym o zadanej długości i nachyleniu	Wypukłe otoczenie
9,0	Przekształcenie afiniczne (rozmiar wyjściowy poszerzony, by wynik był w całości na obrazie; interpolacja)	Filtracja odchylenia standardowego w zadanej masce. Dla RGB std zbiór składa się z potrójnej ilości danych(*). Znormalizować.	9,0	Zamknięcie elementem linijnym o zadanej długości i nachyleniu	Etykietowanie

\* Potrójny zbiór danych oznacza, że daną procedurę wykonujemy na zbiorze powstałym ze sklejenia wartości R,G,B. Oznacza to, że dla maski 5x5 liczymy odchylenie / entropię z 75 wartości (5x5x3) i jako wynik zapisana jest tylko jedna wartość.

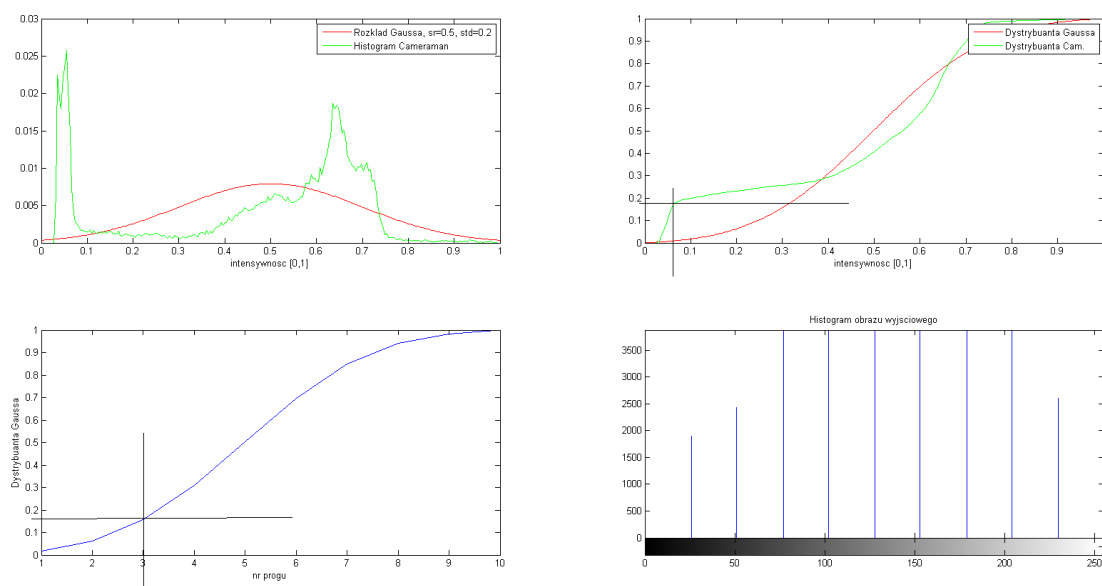
\*\* W przypadku elementów liniowych kąt nachylenia ma być dowolny, a nie tylko wielokrotność 45°.

\*\*\* Normalizacja obrazu (std, entropia, odległość geod. ) oznacza, że w obrazie wyniko-

wym zakres  $\langle \min, \max \rangle$  w obrazie wynikowym ma być rozciągnięty do przedziału  $\langle 0, 255 \rangle$  dla uint8 lub  $\langle 0, 1 \rangle$  dla double.



Rysunek 1: Normalizacja linią łamaną przechodzącą przez 3 punkty.



Rysunek 2: Wyrównanie histogramu do rozkładu Gaussa.

## Przekształcenia punktowe i geometryczne

### 1 Przekształcenia punktowe

Przekształcenia punktowe (bezkontekstowe) są to przekształcenia dotyczące stopnia szarości lub nasycenia barwy dla każdego punktu oddzielnie, dla których nie mają wpływu wartości w punktach sąsiednich.

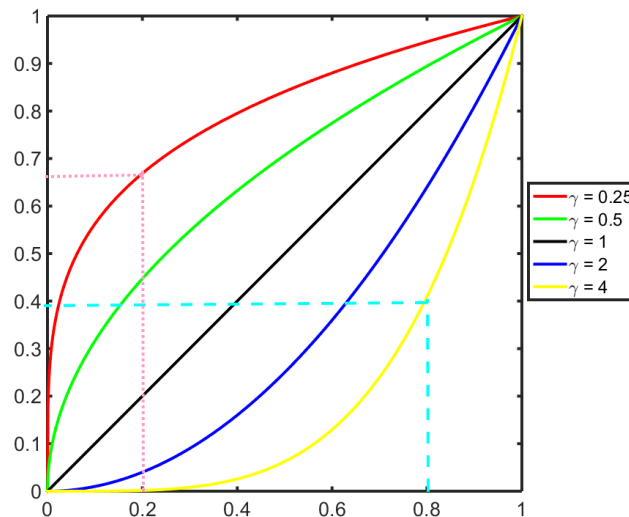
#### 1.1 Liniowe przekształcenia obrazu

Liniowe przekształcenie polega na zmianie wartości piksela przy użyciu przekształcenia liniowego. Zaliczamy do nich między innymi operacje dodawania, negacji czy mnożenia przez pewną liczbę. Służą one np. do rozjaśniania obrazu. Należy pamiętać, że przy stosowaniu kodowania `uint8` dozwolone wartości są z przedziału 0-255, a dla kodowania `double` dozwolone są wartości z przedziału 0-1.

#### 1.2 Nieliniowe przekształcenia obrazu

Do nieliniowych przekształceń zaliczamy np. pierwiastkowanie, potęgowanie logarytmowanie. W niektórych przypadkach jest to znacznie skuteczniejsze niż liniowe korekcje. Szczególne znaczenie ma korekcja gamma  $\gamma$  dana wzorem eq. 1. Idee działania współczynnika  $\gamma$  na obraz przedstawia Rys. 1.

$$Nowy(m, n) = Stary(m, n)^\gamma \quad (1)$$



**Rysunek 1:** Idea działania korekcji  $\gamma$ . Dla  $\gamma < 1$  zwiększony jest kontrast dla "ciemnej" części obrazu kosztem "jasnej" części. Dla  $\gamma > 1$  efekt jest odwrotny: zyskujemy kontrast w części "jasnej" kosztem "ciemnej".

### 1.3 Normalizacja

Normalizacja ma na celu rozciągnięcie zakresu tonalnego obrazu. Jeżeli intensywność punktów na obrazie zajmuje tylko część dozwolonej skali, to możliwe jest rozciągnięcie intensywności tak, by zajmowała pełną skalę. Stosowana jest normalizacja zarówno w wymiarze globalnym, jak i lokalnym. Jeżeli przez  $M$  oznaczamy minimum w normalizowanym obszarze, przez  $N$  maksimum, a przez  $L$  i  $U$  odpowiednio minimum i maksimum skali, to intensywność punktu wynosi:

$$Nowy(m, n) = \left( \frac{L - U}{M - N} \cdot (Stary(m, n) - M) \right) + L \quad (2)$$

Obraz metodą globalną normalizuje się przy użyciu funkcji:

`imadjust(obraz, [low_in; high_in], [low_out; high_out], gamma).`

### 1.4 Binarizacja

Binarizacja to zamiana dowolnego obrazu na obraz logiczny. Stosuje się kilka rodzajów binaryzacji:

- z dolnym progiem: wszystko poniżej progu: 0, powyżej 1

$$\mathbf{L}(\mathbf{m}, \mathbf{n}) = \begin{cases} 0 & \text{gdy } L(m, n) \leq prog \\ 1 & \text{gdy } L(m, n) > prog \end{cases}$$

- z górnym progiem: negatyw binaryzacji z dolnym progiem

$$\mathbf{L}(\mathbf{m}, \mathbf{n}) = \begin{cases} 0 & \text{gdy } L(m, n) > prog \\ 1 & \text{gdy } L(m, n) \leq prog \end{cases}$$

- z dwoma progami: przypomina funkcję bramkową lub jej negatyw
- wielokryterialna: kilka progów i tylko niektóre przedziały mają wartość prawda / fałsz.
- z histerią: wymaga podania 2 wartości progowych ( $prog1 < prog2$ ), wartość piksela  $L(m, n)$  dana jest wzorem:

$$\mathbf{L}(\mathbf{m}, \mathbf{n}) = \begin{cases} 0 & \text{gdy } L(m, n) \leq prog1 \\ s & \text{gdy } prog1 < L(m, n) \leq prog2 \\ 1 & \text{gdy } L(m, n) > prog2 \end{cases}$$

$s$  - wartość sąsiadujących punktów.

- automatyczne progowanie metodą maksymalnej entropii

Wartość entropii  $H$  dla obrazu  $\mathbf{A}$  definiowana jest poprzez eq. 3.

$$H = - \sum_{k=0}^{N_{kolor}} (p(k) \cdot \ln p(k)) \quad (3)$$

gdzie  $p(k)$  - prawdopodobieństwo wystąpienia intensywności  $k$  w obrazie  $\mathbf{A}$ ,  $N_{kolor}$  - maksymalna wartość intensywności występująca w obrazie  $\mathbf{A}$ .

Binarizacja metodą maksymalnej entropii polega na znalezieniu takiej wartości progu  $k$ , dzielącej obraz na tło (piksele o intensywności mniejszej lub równej  $k$ )



i obiekty (o intensywności większej niż  $k$ ), która maksymalizuje funkcję  $H(k)$  (eq. 4). Metoda ta wymaga, by obiekty i tło miały w przybliżeniu podobną powierzchnię.

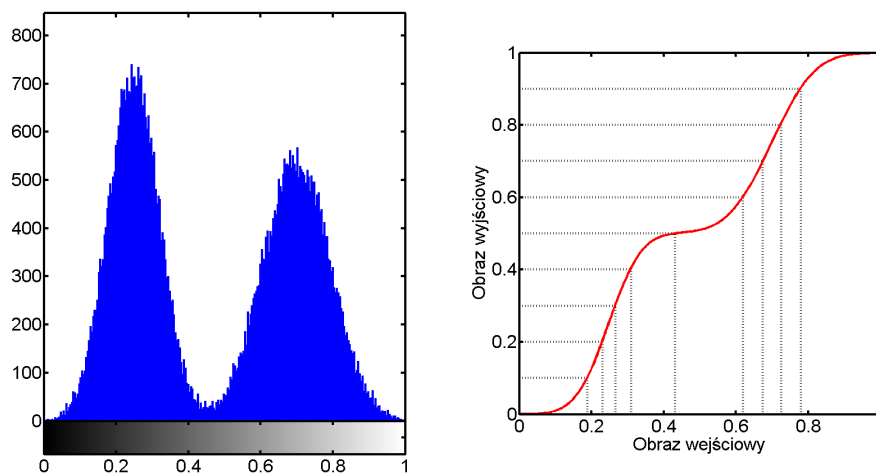
$$H(k) = -\frac{1}{N_k(tlo)} \sum_{i=0}^k (p(i) \cdot \ln p(i)) - \frac{1}{N_k(obiekt)} \sum_{i=k+1}^{N_{kolor}} (p(i) \cdot \ln p(i)) \quad (4)$$

gdzie  $N_k(tlo)$  - ilość elementów tła, a  $N_k(obiekt)$  - ilość pikseli obiektów uzyskanych przy binaryzacji z progiem  $k$ ,  $i$  - intensywność.

## 2 Wyrównywanie histogramu

Histogram wykreśla się poleceniem `imhist(obraz)`. Histogram jest to wykres słupkowy określający ile punktów przybiera daną wartość (intensywność). Wyrównywanie histogramu polega na "upodobnieniu" go do histogramu dla rozkładu równomiernego (wszystkie intensywności mają jednakową częstotliwość występowania) (Rys. 2). Do wyrównywania histogramu służy polecenie `histeq(obraz, ilość_klas)`. Czasami stosuje się wyrównywanie lokalne, tzn. dzieli się obraz na kilka części i każdą wyrównuje oddzielnie. Do tego stosuje się funkcję `adapthisteq(obraz, parametry)`. Do najważniejszych parametrów tego przekształcenia zaliczamy:

- **NumTiles** - Wielkość elementu, dla którego wyrównujemy histogram
- **ClipLimits** - skalarna wielkość decydująca o kontraście (0,1)
- **NBins** - Ilość klas histogramu
- **Range** - zakres wyrównywanych wartości
- **Distribution** - rodzaj histogramu, do którego będzie normowany obraz: 'uniform' (płaski), 'Rayleigh' (dzwonkowy) i 'Exponential' (krzywa eksponenty).
- **Alpha** - parametr rozkładu histogramu, tylko dla Rayleigh i Exponential.



**Rysunek 2:** Idea wyrównywania histogramu obrazu. 1) Liczymy histogram; 2) Liczymy kumulantę (czerwona linia) i dzielimy ją przez jej maksymalną wartość; 3) Dzielimy Oś OY na wykresie kumulanty na "n" równych odcinków (klas histogramu wyjściowego); 4) Szukamy wartości rzutów wartości poszczególnych progów na Oś OX poprzez kumulantę; 5) Wszystkie wartości z OX z ograniczone dwoma rzutami zamieniamy na odpowiadającą im wartość OY.

### 3 Przekształcenia geometryczne

Do przekształceń geometrycznych zaliczamy między innymi: przesunięcie, obrót, zniekształcenie, odbicia symetryczne, powielanie fragmentów. Mają one wpływ na kształt, wielkość lub wygląd obrazu.

#### 3.1 Przesunięcie o wektor

Możliwe są przesunięcia o wektor z dodaniem nowej powierzchni (zwiększenie wymiarów obrazu wyjściowego o wektor) lub przesunięcie w sposób cykliczny z użyciem funkcji `circshift(obraz, wektor)`.

#### 3.2 Obrót

Obraz można obracać przy użyciu funkcji `imrotate(obraz, kąt w stopniach, opcje)`. Pierwszą opcją są metody interpolacji nowych danych:

- najbliższego sąsiada (`'nearest'`);
- dwuliniowa (`'bilinear'`);
- bikubiczna (`'bicubic'`).

Drugą opcją jest rozmiar wyjściowy obrazu:

- `'crop'`: Obraz wyjściowy ma taki sam rozmiar jak wejściowy. Rogi są obcięte, wolne miejsca wypełnione zerami.
- `'loose'`: Obraz wyjściowy na ogół jest powiększony tak, by zmieścił się cały obraz. Wolne miejsca wypełnione zerami.

Opcjami domyślnymi jest `'nearest'`, `'loose'`. Użycie kąta dodatniego powoduje obrót przeciwnie do kierunku wskazówek zegara.

Rotacja obrazu polega na wyliczeniu nowych współrzędnych danego piksela obrazu. Sprowadza się to do wymnożenia wektora współrzędnych przez macierz rotacji:

$$\begin{bmatrix} x^{obr} \\ y^{obr} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (5)$$

#### 3.3 Odbicia symetryczne

Odbicia symetryczne mogą być względem prostej pionowej (`fliplr (obraz)`) lub poziomej (`flipud (obraz)`). Odbicia te (a zwłaszcza odbicie poziome) mają zastosowanie podczas kalibracji (ang. registering) zdjęć (przypisania współrzędnych, np. geograficznych, do poszczególnych pikseli obrazu).

#### 3.4 Dodawanie wierszy, kolumn do obrazu

Dodawanie kolumn i wierszy do obrazu ma na celu głównie podczas filtracji w celu zmniejszenia efektów brzegowych. Służy do tego polecenie `padarray (obraz, rozmiar, metoda, kierunek)`;

Rozmiar w formie wektora `[x y]` mówi, ile wierszy `x` i ile kolumn `y` należy dodać.

Do metod zaliczamy:

- `'circular'` - doklejanie cykliczne

- 'replicate' - doklejanie poprzez kopiowanie skrajnych wiersów / kolumn
- 'symmetric' - odbicie symetryczne.

Istnieją 3 kierunki:

- 'pre': doklejanie z góry i z lewej;
- 'post': z dołu i z prawej;
- 'both': z obu jednocześnie.

### 3.5 Zmiana kształtu i wymiarów obrazu

Zmiana wymiarów obrazu odbywa się przy wykorzystaniu polecenia `reshape (obraz, [nowy rozmiar], 'opcja')`. Do poprawnego działania funkcji musi być spełniony warunek zachowania powierzchni:  $(x^{old} * y^{old} = x^{new} * y^{new})$

Do zmiany kształtu, bez konieczności spełnienia powyższego warunku może służyć polecenie `imresize(obraz, [nowy rozmiar], 'opcja')`.

Zmiana kształtu może odbywać się przy wykorzystaniu dwóch poleceń: `maketform (typ, parametry)` do tworzenia maski przekształcającej i `imtransform (obraz, forma)`; do wykonania samego przekształcenia.

W nowszych wersjach polecenie `maketform` zostało zastąpione poleceniami:

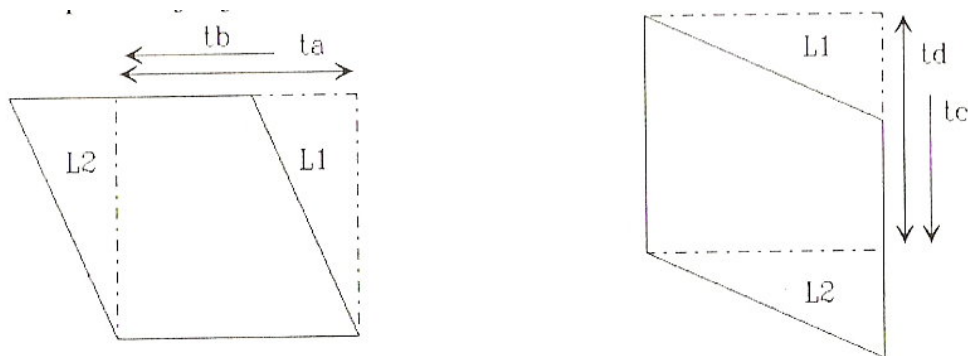
`affine2d([ macierz 3x3 ])` i `projective2d([ macierz 3x3 ])`. W przypadku afinicznym, ostatnia kolumna musi mieć wartości  $[0; 0; 1]$ , a dane w trzecim wierszu nie mają wpływu na obraz.

Polecenie `imtransform` zostało zastąpione natomiast poleceniem `imwarp(obraz, maska)`.

Wyróżniamy następujące typy transformacji:

- ('affine',  $[T_A T_B; T_C T_D; T_E T_F]$ ) - przekształcenie afiniczne: każde różnowartościowe przekształcenie geometryczne, które wszystkie proste zawarte w dziedzinie tego odwzorowania przekształca na proste (Rys. 3). Przekształcenie afiniczne dane jest wzorem:

$$[x_{new} y_{new}] = [x y 1] \cdot \mathbf{T} \quad (6)$$



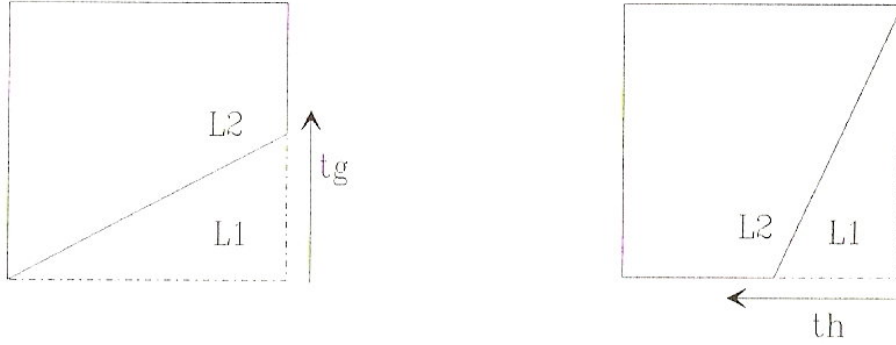
**Rysunek 3:** Parametry przekształcenia afinicznego (źródło:Wróbel & Koprowski, 2004)

- ('projective',  $[T_A T_D T_G; T_B T_E T_H; T_C T_F T_I]$ ): takie przekształcenie, gdzie każde punkty leżące na jednej prostej przechodzą w punkty leżące na drugiej prostej (Rys. 4). Projektcja dane jest wzorem:

$$[x_{new} y_{new} z_{new}] = [x y w] \cdot \mathbf{T} \quad (7)$$

. W przypadku obrazów 2D, powyższe równania można zapisać w następujący sposób:

$$x_{new} = \frac{T_A \cdot x + T_B \cdot y + T_C}{T_G \cdot x + T_H \cdot y + T_I} \quad y_{new} = \frac{T_D \cdot x + T_E \cdot y + T_F}{T_G \cdot x + T_H \cdot y + T_I} \quad (8)$$



**Rysunek 4:** Parametry projekcji (źródło:Wróbel & Koprowski, 2004)

- 'custom'
- 'box'
- 'composite'

### 3.6 Rozpoznanie transformacji geometrycznej

Polecenie `fitgeotrans` pozwala na estymację wartości parametrów transformacji (np. obrotu czy skalowania). W tym celu konieczne jest określenie par punktów na dwóch obrazach (przed i po operacji) odpowiadającym tej samej lokalizacji. Do tej operacji można użyć funkcji `cpselect`. W zależności od ilości wyznaczonych par punktów, możliwa jest estymacja wartości parametrów przekształceń:

- 'Nonreflective similarity' - dla dwóch i więcej par punktów
- 'Affine' - dla 3 i więcej par punktów
- 'projective' - dla 4 i więcej par punktów

### 3.7 Fraktale

Przekształcenia afiniczne można wykorzystać do generowania fraktali (J. Kudrewicz, *Fraktale i Chaos*, WNT, 1993). W tym celu wykonywane są losowo wybrane przekształcenia afiniczne na pojedynczej współrzędnej. Wartość elementu tablicy o obliczonych współrzędnych zostaje zwiększona o jeden. Przykładowy komplet tablic przekształceń afinicznych zamieszczono poniżej.

$$\begin{bmatrix} -0.67 & -0.02 & 0 \\ -0.18 & 0.81 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.4 & 0.4 & 0 \\ -0.1 & 0.4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.4 & -0.4 & 0 \\ -0.1 & 0.4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.1 & 0 & 0 \\ 0.44 & 0.44 & -2 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

## Filtracja w domenie przestrzeni

### 1 Filtracja

Filtracja liniowa jest procesem splotu (konwolucji) obrazu z maską (filtrem). Dla dwuwymiarowej i dyskretniej funkcji filtracja dana jest wzorem:

$$L2(m, n) = (w \times L1)(m, n) = \sum_{p, q \in MF} L1(m - p, n - q)w(p, q), \quad (1)$$

gdzie  $w(p, q)$  - maska filtru MF

Do filtracji służy polecenie `imfilter(obraz, maska, opcje)`; Dostępne opcje są pogrupowane w 3 kategoriach:

- Opcje związane z brzegiem obrazu:
  - **X** - punkty leżące na zewnątrz obszaru, a pokryte maską, mają przypisaną wartość X. Domyślnie X=0;
  - 'symmetric' - potrzebne obszary są tworzone poprzez odbicie symetryczne
  - 'replicate' - potrzebne obszary są tworzone poprzez kopiowanie brzegowych wartości
  - 'circular' - potrzebne obszary są doklejane cyklicznie
- Opcje związane z rozmiarem wyjściowym:
  - 'same' - obraz wyjściowy ma taki sam rozmiar jak wejściowy
  - 'full' - obraz wyjściowy jest większy niż wejściowy
- algorytm filtracji
  - 'corr' - filtracja jest dokonywana z wykorzystaniem funkcji korelacji.
  - 'conv' - wielowymiarowa filtracja z wykorzystaniem splotu.

Poprzez maskę należy rozumieć macierz o wymiarach PxQ (najczęściej kwadratową o nieparzystym wymiarze), której element centralny wskazuje punkt obrazu cyfrowego podlegającemu filtracji.

#### 1.1 Rodzaje filtracji

Filtrację można podzielić na względem kilku kryteriów:

- wg pasma przepuszczania: np. dolno-, górno-, pasmoprzepustowe;
- wg liniowości: liniowe i nieliniowe.
- wg istnienia uprzywilejowanych kierunków działania: izotropowe, anizotropowe

### 2 Filtracja liniowa

#### 2.1 Filtracja dolnoprzepustowa

Przypisują one elementowi centralnemu średnią z obszaru pokrytego maską. Stosuje się do tego maski typu:  $MF = \text{ones}(3)/9$ ,  $MF = \text{ones}(5)/25$ . Działanie tego rodzaju filtru powoduje na ogół rozmycie krawędzi, generalizację obrazu oraz zawężenie zakresu intensywności. Cechą charakterystyczną tej filtracji jest, że suma elementów maski równa się 1. Najczęściej spotykane maski bazują na średniej ważonej, gdzie waga elementu centralnego jest większa od pozostałych, np.  $MF = [1 \ 1 \ 1; \ 1 \ 2 \ 1; \ 1 \ 1 \ 1]/10$ .

## 2.2 Filtracja górnoprzepustowa

Filtracja górnoprzepustowa służy do wyeksponowania detali i zmienności na obrazach, takich jak krawędzie, narożniki, pojedyncze punkty. Wykorzystuje się do tego pierwsze i drugie pochodne intensywności. Najczęściej używanymi operatorami są: gradienty Roberta, Sobela czy Prewitta oraz laplasjany.

### 2.2.1 Operator Prewitta

Operator ten jest oparty o pierwszą pochodną. Wykorzystuje on następujące maski (po lewej maska horyzontalna, po prawej wertykalna):

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

### 2.2.2 Operator Sobela

Operator ten jest oparty o uśrednioną pochodną dyskretną, gdzie wartość pochodnej liczonej przez środek ma dwukrotnie większą wagę niż pochodne liczone z elementów sąsiadujących. Wykorzystuje on następujące maski (po lewej maska horyzontalna, po prawej wertykalna):

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

### 2.2.3 Operator Roberta

Operatory Roberta to osiem masek, obracanych o 45 stopni względem siebie.

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}$$

### 2.2.4 Operator Nevatia-Babu

Operator Nevatia-Babu bazuje na obszarze 5x5 i wyznacza gradienty dla kierunków 0°, 30° i 60° (od lewej do prawej).

$$\begin{bmatrix} 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 & 100 \\ 0 & 0 & 0 & 0 & 0 \\ -100 & -100 & -100 & -100 & -100 \\ -100 & -100 & -100 & -100 & -100 \end{bmatrix} \quad \begin{bmatrix} 100 & 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 78 & -32 \\ 100 & 92 & 0 & -92 & -100 \\ 32 & -78 & -100 & -100 & -100 \\ -100 & -100 & -100 & -100 & -100 \end{bmatrix}$$

$$\begin{bmatrix} 100 & 100 & 100 & -32 & -100 \\ 100 & 100 & 92 & -78 & -100 \\ 100 & 100 & 0 & -100 & -100 \\ 100 & 78 & -92 & -100 & -100 \\ 100 & 32 & -100 & -100 & -100 \end{bmatrix}$$

### 2.2.5 Operator Kirscha

Operator Kirscha polega na przypisaniu elementowi centralnemu wartości maksymalnej z filtracji ośmioma maskami, obróconymi względem siebie o  $45^\circ$ . Pozwala to lepiej wyodrębnić maksymalną amplitudę z ośmiu kierunków filtracji. Pierwsze trzy maski dane są następującymi tablicami:

$$MF^0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad MF^{45} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad MF^{90} = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$$

### 2.2.6 Składanie filtracji

Ze względu na kierunkowość masek (ich anizotropowość), w celu wykrycia wszystkich krawędzi na obrazie konieczne jest składanie wyników filtracji, uzyskanych w wyniku działania ortogonalnych masek. Do najczęściej spotykanych metod składania można zaliczyć:

- wg Normy L2 - wartość filtracji uzyskujemy w wyniku złożenia przy użyciu normy Euklidesowej obrazów po filtracji:

$$Wynik = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (2)$$

- wg Normy Czebyszewa (maksimum) - wartość filtracji uzyskujemy w wyniku złożenia przy użyciu normy maksimum obrazów po filtracji:

$$Wynik = \max\left(\left|\frac{\partial I}{\partial x}\right|, \left|\frac{\partial I}{\partial y}\right|\right) \quad (3)$$

- wg Normy L1 (czasem 1/2 jest pomijana):

$$Wynik = \frac{1}{2} \left( \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right| \right) \quad (4)$$

### 2.2.7 Laplasjany

Lapsjan jest kombinacją drugich pochodnych cząstkowych funkcji  $I(m,n)$ : (wyostrzające, gradientowe):

$$L(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \quad (5)$$

Suma elementów maski laplasjanów może być równa 0 (wykrywanie krawędzi) lub 1 (wyostrzanie obrazu). Jest to filtracja izotropowa. Do najczęściej spotykanych masek laplasjanów zaliczamy:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0.25 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.25 \\ 0.5 & 0.5 & 0 & -0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & -0.75 & -1.5 & -0.75 & 0 & 0.5 \\ 0.5 & -0.5 & -1.5 & -2 & -1.5 & -0.5 & 0.5 \\ 0.5 & 0 & -0.75 & -1.5 & -0.75 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & -0.5 & 0 & 0.5 & 0.5 \\ 0.25 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.25 \end{bmatrix}$$

## 2.3 fspecial

Polecenie `fspecial(typ, parametry)` tworzy maski specjalne. Do parametrów funkcji `fspecial` należą:

- ('average', [rozmiar]) filtr uśredniający. Wartość domyślna to [3 3]. Maska filtru jest liczona według wzoru:

$$h = \text{ones}(N, M) / (N \cdot M) \quad (6)$$

- ('disk', promień) uśredniająca maska kołowa. Promień domyślnie = 5;
- ('gaussian', rozmiar, sigma) dolnoprzepustowy filtr gaussowski o symetrii kołowej. Wartości domyślne to [3 3] i  $\sigma = 0.05$ . Maska filtru **h** jest tworzona w następujący sposób:

$$h_g(n_1, n_2) = \exp \frac{-(n_1^2 + n_2^2)}{2\sigma^2} \quad (7)$$

$$h(n_1, n_2) = \frac{h_g(n_1, n_2)}{\sum_{n_1} \sum_{n_2} h_g(n_1, n_2)} \quad (8)$$

Zamiast tworzyć maskę i następnie filtrować, można skorzystać z funkcji `imgaussfilt`.

- ('laplacian', alpha) dwuwymiarowy laplasjan,  $\alpha \in (0, 1)$ , domyślnie 0.2. Maska jest tworzona w następujący sposób:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \approx \frac{1}{\alpha + 1} \begin{bmatrix} \alpha & 1 - \alpha & \alpha \\ 1 - \alpha & -4 & 1 - \alpha \\ \alpha & 1 - \alpha & \alpha \end{bmatrix}$$

- ('log', rozmiar, sigma) laplasjan filtru gaussowskiego. Rozmiar domyślny to [5 5], a  $\sigma = 0.5$ . Filtr **h** jest tworzony z wykorzystaniem wzoru 7:

$$h(n_1, n_2) = \frac{(n_1^2 + n_2^2 - 2\sigma^2)h_g(n_1, n_2)}{2\pi\sigma^6 \sum_{n_1} \sum_{n_2} h_g} \quad (9)$$

- ('motion', len, theta) aproksymacja przesuwania się kamery. Len określa długość przesunięcia (9) pod kątem  $\theta$  (domyślnie kąt=0)
- 'prewitt', 'sobel' - horyzontalna maska Prewitta, Sobela
- ('unsharp', alpha) - filtr rozmywający, działanie odwrotne do laplasjanu. Filtr jest tworzony na podstawie wzoru:

$$h = \frac{1}{\alpha + 1} \begin{bmatrix} -\alpha & \alpha - 1 & -\alpha \\ \alpha - 1 & \alpha + 5 & \alpha - 1 \\ -\alpha & \alpha - 1 & -\alpha \end{bmatrix}$$

## 2.4 Operatory krawędziowe kolorowe

Filtracja realizowana jest jako suma różnic trzech par punktów leżących symetrycznie względem punktu centralnego. Różnicę dwóch D punktów P1(R,G,B) i P2(R,G,B) możemy zdefiniować na 4 sposoby<sup>1</sup>:

<sup>1</sup>Opis kolorowych operatorów krawędziowych oraz działania filtru Canny pochodzi z: Kasprzak, W. (2009), *Rozpoznawanie obrazów i sygnałów mowy*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa



1. z wykorzystaniem normy Euklidesowej:

$$D(P1, P2) = \sqrt{(R1 - R2)^2 + (G1 - G2)^2 + (B1 - B2)^2} \quad (10)$$

2. z wykorzystaniem normy L1:

$$D(P1, P2) = |R1 - R2| + |G1 - G2| + |B1 - B2| \quad (11)$$

3. jako maksimum z wartości bezwzględnych poszczególnych składowych:

$$D(P1, P2) = \max(|R1 - R2|, |G1 - G2|, |B1 - B2|) \quad (12)$$

4. jako średnia ważona wartości bezwzględnych różnic poszczególnych składowych:

$$D(P1, P2) = \omega_1 \cdot |R1 - R2| + \omega_2 \cdot |G1 - G2| + \omega_3 \cdot |B1 - B2| \quad (13)$$

### 3 Filtracja nieliniowa

Istnieje również cała gama filtrów, których działanie zależy od np. uporządkowania obrazu. Są to np. filtry medianowe, maksymalizujące, minimalizujące czy entropii, które przypisują filtrowanemu punktowi odpowiednio medianę, maksimum, minimum i entropię z obszaru pokrytego maską. W pakiecie MatLAB 2014b zostały zdefiniowane następujące metody filtracji:

- `medfilt2(obraz, [rozmiar]);` - filtracja medianowa
- `ordfilt2(obraz, nr, zasięg);` - zwraca wartość znajdującą się na pozycji `nr` w uporządkowanym wektorze (1 - minimum, wielkość maski - maksimum).
- `rangefilt(obraz, zasięg)` - zwraca różnicę pomiędzy maksimum i minimum intensywności dla pikseli otoczenia zdefiniowanych w tablicy `zasięg`, np. `ones(3)`.
- `entropyfilt(obraz, zasięg);` - zwraca wartość entropii  $E$  w blokach o zadanym rozmiarze (domyślnie 9x9 pikseli), obliczoną wg wzoru:

$$E = - \sum_{n=0}^N \log p(n) \cdot p(n) \quad (14)$$

$N$  - ilość poziomów intensywności

$p(n)$  - prawdopodobieństwo wystąpienia piksela o intensywności  $n$

- `stdfilt(obraz, zasięg)` - dla każdego piksela zwraca wartość odchylenia standardowego z otoczenia.
- `modelfilt(obraz, [rozmiar])` - filtracja dominanty.

#### 3.1 Operacje blokowe

Inną możliwością tworzenia filtrów nieliniowych są tzw. operacje blokowe. Do implementacji tego typu filtrów służy polecenie `nlfilter(obraz, [rozmiar], funkcja)`, np.: `nlfilter(obraz, [3 3], 'min(x(:))')`.

Cechują się one znacznie dłuższym czasem działania niż gotowe funkcje, dlatego jeżeli istnieją gotowe funkcje filtrujące (jak np. `medfilt2`), należy z nich korzystać, a operacje blokowe stosować tylko w ostateczności.

### 3.2 Funkcja edge

Do wykrywania naroży może również służyć funkcja `edge(obraz, 'metoda', parametry)`. Funkcja ta na wejściu przyjmuje obraz monochromatyczny, a zwraca mapę logiczną, gdzie wartość logiczną "1" ma wykryty narożnik. Funkcja ta posiada 6 metod detekcji krawędzi (zarówno liniowych, jak i nieliniowych):

- `edge(obraz, 'Sobel', próg, 'kierunek', 'opcje')`. Wykorzystuje ona maskę Sobela. Dozwolone są następujące kierunki: 'horizontal', 'vertical', 'both'. Do opcji zaliczamy operację 'thinning' i 'nothinning'
- `edge(obraz, 'prewitt', próg, 'kierunek')`. Operator Prewitta, parametry analogicznie do Sobela.
- `edge(obraz, 'roberts', próg, 'kierunek', 'opcje')` - filtracja wykorzystująca operator Robertsa.
- `edge(obraz, 'log', próg, sigma)` - Laplasjan metody Gaussa. Parametr sigma określa standardowe odchylenie metody. Wartością domyślną jest 2. Wielkość maski filtru ( $n \times n$ ) jest liczona według wzoru:  $n = \text{ceil}(\sigma * 3) * 2 + 1$
- `edge(obraz, 'zerocross', próg, filtr_h)`
- `edge(obraz, 'canny', [ $\theta_1, \theta_2$ ], sigma)` - operator Canny.  $\theta_1$  określa dolny próg odcięcia,  $\theta_2$  określa górny. Jeżeli podany jest tylko jeden próg, to jest on traktowany jako górny, a dolny jest liczony według wzoru:  $\theta_1 = 0.4 \cdot \theta_2$ . Sigma jest to standardowe odchylenie dla filtru gaussowskiego. Domyślnie  $\sigma = 1$ . Filtr Canny działa w kilku krokach:

1. Wygładzanie obrazu za pomocą filtru Gaussa o wariancji  $\sigma^2$
2. Wyznaczenie obrazu krawędziowego prostym operatorem krawędziowym w obszarze 2x2 lub 3x3
3. Cieniowanie krawędzi metodą tłumienia niemaksymalnego - zerowanie elementów krawędziowych niebędących maksimumami lokalnymi
4. Cieniowanie krawędzi metodą tłumienia z progami histerezy  $\theta_1, \theta_2$

Dla wszystkich tych metod, poleceniem `[BW, próg]=edge(...)` można uzyskać informacje o użytym progu detekcji krawędzi. Dla metod gradientowych (Sobel, Prewitt, Roberts) automatyczny próg binaryzacji jest liczony na podstawie magnitudy policzonego gradientu.

### 3.3 VMF - Vector median flow

Jest to filtracja stosowana wyłącznie dla obrazów wielokanałowych (np. RGB, CMYK, satelitarne). Polega ona na zastąpieniu wartości piksela centralnego przez wartość koloru leżącego najbliżej w przestrzeni kolorów. Jeżeli w masce istnieje piksel o identycznej wartości koloru jak piksel centralny, to wartość piksela nie ulega zmianie. Odległość dwóch wartości kolorów w danej przestrzeni najczęściej liczona jest z użyciem normy L1 lub L2.

## 4 Filtracja adaptacyjna

Osobną kategorią filtrów nieliniowych są filtry adaptacyjne. Są to filtry, których charakterystyka zmienia się w zależności od analizowanego obszaru.

Dolnoprzepustowa **filtracja Wienera** działa dwuetapowo:

1. Dla każdego punktu i jego otoczenia obliczamy wartość parametru, który kwalifikuje dany punkt jako należący lub nie do krawędzi.
2. jeżeli dany punkt został zakwalifikowany jako nie należący do krawędzi, zostaje on poddany silnemu uśrednieniu. W przeciwnym wypadku jego wartość pozostaje bez zmian lub poddany zostaje uśrednieniu o niewielkiej mocy.

Realizujemy ją poleceniem: `wiener2(obraz, [rozmiar])`.

Innym rodzajem jest **adaptacyjny filtr medianowy**. Działa on w następujący sposób:

1. Dla każdego piksela obliczamy średnią i odchylenie standardowe w pewnej masce.
2. Jeżeli wartość intensywności piksela centralnego mieści się w zakresie  $\text{średnia} \pm (\text{współczynnik} \cdot \text{odchylenie})$ , to wartość piksela nie ulega zmianie.
3. Jeżeli wartość piksela jest poza przedziałem, to zastępowana jest on wartością medianą z wartości pikseli otoczenia mieszczących się we wspomnianym przedziale.

Ilość zmian zależy od wielkości maski oraz od wartości współczynnika. Jego wartość w większości przypadków nie przekracza wartości 2 (zwykle wynosi 1).

## 5 Dekonwolucja

Operacją odwrotną do splotu (czyli m.in. filtracji) jest dekonwolucja. Istnieje kilka technik dekonwolucji zaimplementowanych w MatLABie:

- Dekonwolucja ślepa: `deconvblind`
- Dekonwolucja przy użyciu algorytmu Lucy-Richardsona: `deconvlucy`
- Dekonwolucja przy użyciu algorytmu Wienera: `deconvwnr`
- Dekonwolucja "unormowana": `deconvreg`

## 6 Szum

Do "zaszumiania" obrazów służy polecenie `imnoise(obraz, typ, parametry)`. Podstawowymi rodzajami szumu są:

- 'gaussian' - szum o rozkładzie gaussa z średnią  $m$  i wariancją  $\sigma^2$ . Domyślnie  $m=0$  i  $\sigma^2=0.01$ ;
- 'localvar' - szum gaussowski o średniej  $m=0$  i wariancji zależnej od otoczenia punktu. Aby użyć tej funkcji należy jako parametr dodać tablicę rozkładu wariancji o rozmiarze równym zaszumionemu obrazowi.
- 'poisson' - szum o rozkładzie Poissona
- 'salt & pepper' - (szum impulsowy) zamienia wartość piksela na minimum lub maksimum z dostępnej skali z gęstością zmiany  $d$ . Domyślnie  $d=0.05$ ;
- 'speckle' - multiplikatywny szum zmieniający wartość piksela zgodnie z równaniem:

$$L(m, n) = L(m, n) + n \cdot L(m, n) \quad (15)$$

gdzie  $n$  - liczba z rozkładu normalnego o średniej  $m=0$  i wariancji  $\sigma^2$ . Domyślnie  $\sigma^2=0.04$ ;

## Przekształcenia morfologiczne

### 1 Element strukturalny

Przekształcenia morfologiczne są to operacje związane z zastosowaniem tzw. Elementu strukturalnego (SE). Jest to pewien wycinek obrazu z wyróżnionym jednym punktem (elementem centralnym). Element strukturalny jest przemieszczany po całym obrazie i dla każdego punktu wykonywana jest analiza koincydencji punktów obrazu i elementu strukturalnego.

Do tworzenia elementu strukturalnego służy polecenie `strel( 'kształt', parametry );`. Możliwe są następujące kształty:

- ('arbitrary', NHOOD); - dowolny element definiowany poprzez macierz NHOOD;
- ('pair', Offset) - zawiera dwa elementy oddalone o offset
- ('diamond', R) - kwadrat obrócony o 45°. R - połowa przekątnej
- ('periodicline', P, V) - (2\*P+1) punktów wzdłuż linii oddalonych o wektor V między sobą.
- ('disk', R) - koło o promieniu R
- ('rectangle', [M N]) - prostokąt wypełniony jedynkami
- ('line', Len, Deg) - linia o długości Len i nachyleniu Deg
- ('square', w) - kwadrat o boku w
- ('octagon', R) - ośmiokąt o promieniu R

### 2 Erozja

Aby zdefiniować erozję należy założyć, że istnieje nieregularny obszar X i koło B o promieniu r. Zerodowana figura X elementem B jest to zbiór wszystkich środków kół o promieniu r, które w całości zawarte są wewnątrz obszaru X. W implementacji komputerowej polega to na usunięciu wszystkich elementów o wartości 1, które posiadają choć jednego sąsiada o wartości 0. Można to interpretować jako filtr minimalny. Do erozji służy polecenie `imerode(obraz, SE)`. Jako elementów strukturalnych używa najczęściej następujących masek:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & X & X \\ X & 1 & X \\ X & X & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

gdzie X - dowolna wartość obrazu {0,1}

W MATLABie poszczególne elementy koduje się następująco:

1 to 1; 0 to -1; X to 0.

#### 2.1 Cechy erozji

Do najważniejszych cech erozji należą:

- w większości przypadków jest addytywna, tzn. erozję o założonej wielkości można interpretować jako złożenie odpowiedniej ilości erozji o mniejszej wielkości. Twierdzenie to jest nieprawdziwe przy bardzo małych elementach strukturalnych, których kształt istotnie odbiega od teoretycznego elementu strukturalnego;

- erozja złożonym SE jest równoważna złożeniu erozji poszczególnymi elementami tego SE;
- położenie punktu centralnego SE nie ma większego znaczenia. Zmiana położenia punktu centralnego o dany wektor powoduje przesunięcie obrazu wynikowego o ten sam wektor;
- erozja ma zdolność do eliminacji drobnych szczegółów i wygładzania brzegu figury;
- erozja elementem o podłużnym kształcie pozwala uwypuklić linijne fragmenty zorientowane w tym samym kierunku, co SE;
- erozja dokonuje generalizacji obrazu.

### 3 Dylatacja

Jest to przekształcenie odwrotne do erozji. Zachowując założenia z erozji, figura po dylatacji jest zbiorem środków wszystkich kół B, dla których choć jeden ich punkt pokrywa się z jakimkolwiek punktem figury X. Analogicznie do erozji, można to przekształcenie traktować jako filtr maksymalizujący. Dylatację można dodatkowo zdefiniować jako negatyw erozji negatywu obrazu. Do dylatacji służy polecenie `imdilate(obraz, SE)`; Typowy element strukturalny dylatacji wygląda następująco:

$$\begin{bmatrix} X & X & X \\ X & 0 & X \\ X & X & X \end{bmatrix}$$

Przy czym przynajmniej jeden element X musi mieć wartość 1.

#### 3.1 Cechy Dylatacji

Do najważniejszych cech dylatacji należą:

- zamykanie małych otworów i wąskich zatok
- zdolność do łączenia obiektów blisko połączonych
- podobnie jak w przypadku erozji addytywność i możliwość zamiany dylatacji złożonym SE na dylatację elementami składowymi.

### 4 Otwarcie i Zamknięcie

Są to operacje będące złożeniem erozji i dylatacji. Otwarcie polega na następujących po sobie erozji i dylatacji tym samym elementem, a zamknięcie następującej po sobie dylatacji i erozji:

$$\begin{aligned} \text{otwarcie} &= \text{dylatacja}(\text{erozja}, \text{SE}), \text{SE}) \\ \text{zamknięcie} &= \text{erozja}(\text{dylatacja}, \text{SE}), \text{SE}) \end{aligned}$$

Do wykonania tych operacji służą następujące polecenia:  
`imopen(obraz, SE);` i `imclose(obraz, SE);`.

Otwarcie powoduje usunięcie drobnych obiektów takich jak np. półwyspy, wypustki, nie zmieniając zasadniczo wielkości figury. Może rozłączyć niektóre obiekty połączone pierwotnie przewężeniami. Zamknięcie wypełnia wąskie zatoki oraz drobne otwory wewnątrz figury. Może też połączyć leżące blisko siebie obiekty.

## 4.1 Cechy Otwarcia i Zamknięcia

Najważniejszymi cechami Otwarcia  $O(F)$  i Zamknięcia  $C(F)$  są:

- Przekształcenia te są niezmiennie wobec siebie:  $O(O(F))=O(F)$  i  $C(C(F))=C(F)$
- Otwarcie usuwa drobne obiekty i szczegóły, może też rozłączyć niektóre obiekty z przewężeniami
- Zamknięcie wypełnia wąskie zatoki i wcięcia oraz drobne otwory wewnątrz obiektu. Może też połączyć leżące blisko siebie elementy.
- Zależność pomiędzy polem powierzchni figury  $F$ , a wynikiem Eroзии, Otwarcia, Zamknięcia i Dylatacji tej figury jest następująca:

$$E(F) \leq O(F) \leq C(F) \leq D(F) \quad (1)$$

## 5 Gradient morfologiczny

Gradient morfologiczny jest definiowany na trzy sposoby:

- jako różnica między obrazem wejściowym, a wynikiem jego eroзии;
- jako różnica między wynikiem dylatacji, a obrazem wejściowym;
- jako różnica między wynikiem dylatacji obrazu wejściowego, a jego erozią. W tym wariancie wynik często bywa dzielony przez 2.

## 6 Rekonstrukcja

Rekonstrukcja obrazu przy użyciu markera. Obraz i marker mogą być obrazami logicznymi lub monochromatycznymi i muszą mieć ten sam rozmiar. Rekonstrukcja polega na cyklicznym dokonywaniu dylatacji obrazu i wyznaczaniu części wspólnej obrazu uzyskanego po dylatacji i obrazu wejściowego dla całego przekształcenia. Cykl ten powtarza się aż do uzyskania zbieżności, to znaczy braku zmian pomiędzy sąsiednimi iteracjami.

W MatLABie używane jest do tego polecenie:

`imreconstruct(marker, obraz, sąsiedztwo).`

## 7 Operacja Hit or Miss

Operacja Hit or miss (trafi, nie trafi) jest definiowane jako:

"Do każdego punktu analizowanego obrazu przykładany jest punkt centralny danego elementu strukturalnego. Jeżeli lokalne otoczenie analizowanego punktu zgodne jest z elementem strukturalnym - odpowiedni punkt obrazu wynikowego uzyskuje wartość 1. W przeciwnym przypadku wartość 0." (Tadeusiewicz & Kohoroda). Stosuje się różne elementy strukturalne, mające różne zastosowanie. I tak:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ X & X & X \end{bmatrix} \quad \begin{bmatrix} X & 1 & X \\ X & 1 & X \\ 1 & X & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & X & X \\ X & 1 & X \\ 1 & X & 1 \end{bmatrix}$$

Pierwszy element służy do detekcji pojedynczych izolowanych punktów, drugi do detekcji punktów końcowych, a pozostałe dwa do detekcji punktów węzłowych.

Do użycia tego przekształcenia służy polecenie `bwhitmiss(obraz, SE1, SE2)` lub `W=bwhitmiss(A, SE);`

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad SE = \begin{bmatrix} 1 & 0 & -1 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \quad W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

## 8 Wypukłe otoczenie

Wypukłe otoczenie jest to najmniejsza figura wypukła zawierająca daną figurę. Realizuje się to przez pogrubienie maską, obracaną co  $45^\circ$ :

$$\begin{bmatrix} 1 & 1 & X \\ 1 & 0 & X \\ 1 & X & 0 \end{bmatrix}$$

## Przekształcenia morfologiczne II i operacje na obrazach logicznych

### 1 Przekształcenia morfologiczne

#### 1.1 Ścienianie i Pogrubianie

Pogrubianie i ścienianie: Operacje te polegają na nałożeniu lub ściągnięciu wierzchniej warstwy obiektu. W procesie ścieniania wartość punktu nie zmienia się, gdy SE nie pokrywa się z jego sąsiedztwem lub zmienia jego wartość na zero, gdy SE pasuje do sąsiedztwa rozpatrywanego punktu. W procesie pogrubiania relacje są odwrotne. Do wykonywania tych operacji służy polecenie `bwmorph` z parametrami `'thicken'` i `'thin'`: `bwmorph(obraz, 'thin')`;

Elementem wykorzystywanym do ścieniania jest maska, obracana co 90°:

$$\begin{bmatrix} X & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

#### 1.2 Szkieletyzacja

Szkielet figury to zbiór wszystkich wszystkich środków okręgów, mieszczących się w całości wewnątrz figury i mających co najmniej dwa punkty wspólne z brzegiem figury. Szkieletyzację wykonuje się przy użyciu następujących elementów:

$$\begin{bmatrix} 1 & 1 & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & X & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & X \\ 1 & 1 & 0 \\ 1 & 1 & X \end{bmatrix}$$

Do szkieletyzacji służy polecenie `bwmorph(obraz, 'skel')`;

#### 1.3 Dylatacja bez stykania obszarów (SKIZ)

SKIZ (ang. skeleton by influence zone) jest pomocne przy rozdzielaniu elementów częściowo się stykających. Jako pierwszy krok wykonuje się erozję, a następnie pogrubienie następującym SE:

$$\begin{bmatrix} X & X & X \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Gdy przekształcenie to jest stosowane cyklicznie, aż do braku zmian, uzyskuje się szkielet wpływów (SKIZ). Strefa wpływów danego punktu definiowana jest jako zbiór wszystkich punktów obrazu, dla których odległość do danego punktu jest mniejsza niż do pozostałych. Przekształcenie to czasami powoduje pewne artefakty (wąskie i głębokie zatoki), które eliminuje się w 3 krokach: negatyw  $\rightarrow$  obcięcie gałęzi  $\rightarrow$  negatyw.



## 1.4 Obcinanie gałęzi

Do obcinania gałęzi, powstałych np. w wyniku szkieletyzacji lub wycieniania służą następujące elementy strukturalne:

$$\begin{bmatrix} X & 0 & 0 \\ 1 & 1 & 0 \\ X & 1 & X \end{bmatrix} \quad \begin{bmatrix} 0 & X & X \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

## 1.5 Pakiet bwmorph

`bwmorph(obraz, 'metoda', parametr)`; Funkcja ta aplikuje operacje morfologiczne na obrazie logicznym. Do metod zaliczamy:

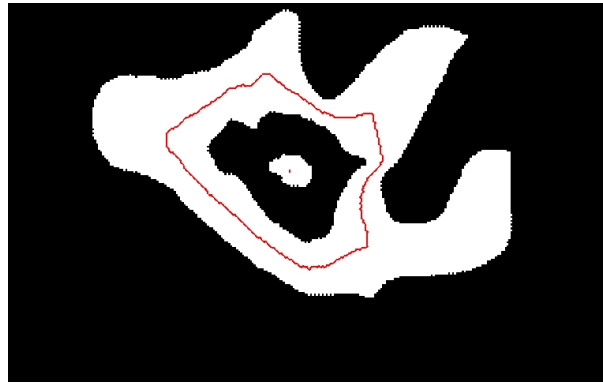
- `'bothat'` - operacja Bottom hat, która polega na odjęciu od zamknięcia obrazu samego obrazu. Istnieje jej uogólnienie na obrazy monochromatyczne `imbothat`.
- `'bridge'` - łączy blisko siebie leżące piksele.
- `'clean'` - zeruje pojedynczego piksela, otoczonego samymi zerami.
- `'close'` i `'open'` - zamknięcie i otwarcie elementem `ones(3)`.
- `'diag'` - zamienia połączenia diagonalne (8-sąsiedztwo) na 4-sąsiedztwo poprzez zmianę wartości piksela leżącego pod przekątną na 1.
- `'dilate'` i `'erode'` - dylatacja i erozja elementem `ones(3)`
- `'fill'` - zmienia wartość zerowego piksela otoczonego samymi jedynkami na jeden.
- `'hbreak'` - rozdziela H-połączone piksele (w pionie i poziomie)

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

- `'majority'` - jeżeli suma maski 3x3 jest większa lub równa 5, to element centralny przyjmuje wartość 1. Jeżeli nie, to element centralny ma wartość 0.
- `'remove'` - zmienia wartość środkowego piksela na 0, jeżeli wszyscy jego 4-sąsiedzi mają wartość 1.
- `('shrink',n)` - kurczy obiekty do punktów. W przypadku obiektu z "dziurami", i użycia tej funkcji z `n=inf`, powstaje obwódka w pomiędzy granicą zewnętrzną i wewnętrzną obiektu. Obiekty jednorodne są zmniejszane do punktu. Parametr `n` oznacza ilość iteracji, przy `n=inf` operacja jest wykonywana do momentu, kiedy pomiędzy wynikami dwóch sąsiednich iteracji nie będzie różnicy. Efekt działania tej funkcji z parametrem `n=inf` (kolor czerwony) naniesiony na obraz wejściowy przedstawia fig.1. Widoczna jest obwódka wokół dziury oraz pojedynczy punkt w wewnętrznym obiekcie.
- `('skel',n)` - szkieletyzacja. Działanie parametru `n` jest analogiczne do funkcji `'shrink'`.
- `'spur'` - usuwa gałęzie, połączone diagonalnie.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- `('thicken', n)` i `('thin', n)` - pogrubianie i wycienianie obiektu



**Rysunek 1:** Efekt działania funkcji `bwboundaries` zaznaczony kolorem czerwonym

- **'tophat'** - operacja Top hat. Funkcja zwraca różnicę między obrazem, a jego otwarciem. Istnieje jej uogólnienie na obrazy monochromatyczne `imtophat`.

## 1.6 Operacje geodezyjne

Operacje geodezyjne wymagają użycia dwóch obrazów (obraz wejściowy oraz maska / marker). W wyniku tych operacji obraz wejściowy jest poddawany operacji morfologicznej, a następnie porównywany z maską. Wyróżniamy następujące operacje geodezyjne:

- **dylatacja geodezyjna (o rozmiarze 1)** - jest to wartość minimalna z obrazu wejściowego poddanego dylatacji jednostkowym elementem strukturalnym oraz maski. W przypadku obrazów logicznych jest to iloczyn logiczny maski oraz obrazu po dylatacji. Zakłada się, że  $D(\text{obraz}) \leq \text{maska}$
- **erozja geodezyjna (o rozmiarze 1)** - jest to wartość maksymalna z obrazu wejściowego poddanego erozji jednostkowym elementem strukturalnym oraz maski. W przypadku obrazów logicznych jest to suma logiczna maski oraz obrazu po erozji. Zakłada się, że  $E(\text{obraz}) \geq \text{maska}$
- **rekonstrukcja przez dylatację** - wykonywanie cyklicznie dylatacji geodezyjnych, aż do braku różnic pomiędzy kolejnymi dylatacjami. Jest to najczęściej stosowana rekonstrukcja morfologiczna.
- **rekonstrukcja przez erozję** - wykonywanie cyklicznie erozji geodezyjnych, aż do uzyskania braku różnicy pomiędzy kolejnymi iteracjami.
- **otwarcie przez rekonstrukcję** - Polega na wykonaniu erozji zadany elementem strukturalnym, a następnie rekonstrukcji przez dylatację.
- **zamknięcie przez rekonstrukcję** - Polega na wykonaniu dylatacji zadany elementem strukturalnym, a następnie rekonstrukcji przez erozję.

## 2 Operacje na obrazach logicznych

Do innych operacji (niż morfologiczne) na obrazach logicznych należą następujące przekształcenia i operacje:

## 2.1 Estymacja pola powierzchni

Polecenie `bwarea(obraz)`; podaje estymowane pole powierzchni. Wartość uzyskana tą funkcją może być inna od wartości rzeczywistej (uzyskanej funkcją `sum(obraz(:))`). Zliczania dokonuje się w otoczeniu 2x2 i na podstawie następujących warunków:

- 0 - jeżeli w otoczeniu wszystkie piksele mają wartość 0.
- 0.25 - jeżeli tylko jeden piksel ma wartość 1.
- 0.5 - jeżeli dwa piksele mają wartość 1 i stykają się bokami.
- 0.75 - jeżeli dwa piksele mają wartość 1 i stykają się rogami.
- 0.875 - jeżeli 3 piksele mają wartość 1.
- 1 - jeżeli wszystkie 4 piksele mają wartość 1.

## 2.2 Wyznaczanie obwodu

Polecenie `bwperim(obraz, sąsiedztwo)`; wyznacza linie obwodu. Sąsiedztwo dla obrazów 2D może przyjmować wartość 4 lub 8. Rozbudowaniem tej funkcji jest polecenie

`D=bwboundaries(obraz, sąsiedztwo, opcje)`.

Jako rezultat działania funkcji otrzymujemy wektor o rozmiarze równym ilości izolowanych obiektów, składający się z tablic współrzędnych każdej granicy. Jako opcje mamy dwie możliwości: `'noholes'` i `'holes'`. Na fig.2 pokazano efekt działania tych parametrów. Kolorem czerwonym zaznaczono granice obliczone z parametrem `'noholes'`. Dzięki zastosowaniu opcji `'holes'` zyskujemy, oprócz dwóch czerwonych, dodatkową granicę zaznaczoną kolorem niebieskim.



**Rysunek 2:** Efekt działania funkcji `bwboundaries` z parametrami `'noholes'` - kolor czerwony, i `'holes'` - czerwony + niebieski

## 2.3 Obliczanie odległości

Polecenie `bwdist(obraz, 'metryka')`; podaje odległość od najbliższego piksela o wartości 1, przy wykorzystaniu jednej z następujących metryk:

- `'euclidean'`:  $d = \sqrt{(m_1 - m_2)^2 + (n_1 - n_2)^2}$
- `'quasi-euclidean'`:

$$d = \begin{cases} |m_1 - m_2| + (\sqrt{2} - 1)|n_1 - n_2| & \text{dla } |m_1 - m_2| > |n_1 - n_2| \\ (\sqrt{2} - 1)|m_1 - m_2| + |n_1 - n_2| & \text{dla } |m_1 - m_2| \leq |n_1 - n_2| \end{cases}$$

- 'chessboard':  $d = \max(|m_1 - m_2|, |n_1 - n_2|)$
- 'cityblock':  $d = |m_1 - m_2| + |n_1 - n_2|$

## 2.4 Liczba Eulera

`bweuler(obraz, sąsiedztwo)`; - podaje wartość liczby Eulera. Jest to ilość obiektów występujących na obrazie binarnym, pomniejszona o ilość dziur w tych obiektach. Sąsiedztwo przyjmuje wartość 4 lub 8.

## 2.5 Wypełnianie dziur w obiektach

Do wypełniania dziur w obiektach służy polecenie:

```
imfill(obraz, sąsiedztwo, lokalizacja, 'holes');
```

We wcześniejszych wersjach biblioteki jest to funkcja `bwfill`. W przypadku obrazów logicznych wypełnia wskazane myszką lub poprzez podanie współrzędnych punktu obszaru złożonego z zer wewnątrz figury. Dodanie parametru 'holes' wypełnia wszystkie dziury na obrazie. W przypadku obrazu monochromatycznego poprzez dziurę rozumie się obszary zbudowane z pikseli o mniejszej intensywności niż otoczenie.

## 2.6 Etykietowanie i segmentacja

Polecenie `bwlabel(obraz, sąsiedztwo)`; służy do etykietowania, tzn. do przypisywania jednakowej wartości pikselom wewnątrz obszarów jednorodnych i rozłącznych z innymi obiektami. Sąsiedztwo przyjmuje wartość 4 lub 8. Każdy jednorodny obiekt ma unikalny numer, będący kolejnymi dodatnimi liczbami całkowitymi.

Do prezentacji wyników etykietowania służy polecenie `label2rgb (wynik, 'paleta', [kolor zerowy], 'kolejność')`. Do etykietowania obrazów wielowymiarowych służy polecenie `bwlabeln (obraz, sąsiedztwo)`.

Z pojęciem etykietowania związana jest segmentacja czyli proces podziału obrazu na fragmenty odpowiadające widocznym na obrazie obiektom. Wyróżniamy dwie grupy metod segmentacji:

- poprzez podział obszaru: polega na stopniowym, iteracyjnym, podziale dużych obszarów na mniejsze, których piksele mają wartość różniącą się od sąsiedztwa.
- poprzez rozrost obszaru: piksele sąsiedztwa obszaru są sprawdzane czy spełniają warunki podobieństwa i w przypadku spełnienia, są dołączane do obszaru.

## 2.7 Wskazywanie obiektów

Polecenie `bwselect(obraz, x, y, sąsiedztwo)`; służy do wskazywania obszarów jednorodnych poprzez wskazanie dowolnego punktu wewnętrznego. Wektory x,y wskazują położenie pikseli wewnętrznych. W przypadku ich braku, wskazywanie odbywa się przy użyciu myszki. Sąsiedztwo=4,8 i oznacza typ sąsiedztwa. Wynikiem działania tego polecenia jest mapa logiczna, gdzie wartość jeden mają tylko obiekty wskazane poprzez punkty wewnętrzne.

## 2.8 Dział wodny

Dział wodny jest to linia rozdzielająca dwa obszary przyciągania (zlewnie). Służy do tego polecenie `watershed(obraz, sąsiedztwo)`; . Funkcja działu wodnego może służyć do rozdzielania blisko siebie leżących obiektów.

## Wprowadzenie do analizy obrazu<sup>1</sup>

Analiza obrazu polega na wydobyciu cech opisujących obraz. W jej skład wchodzi następujące procesy: segmentacja, lokalizacja obiektów oraz wyznaczenie ich cech.

### 1 Schemat analizy

Proces analizy obrazu jest ostatnim etapem całego procesu wydobywania informacji. Schemat na ogół wygląda następująco:

1. Akwizycja obrazu
2. Wstępne przetwarzanie (w większości przypadków jest to ogół operacji na obrazach kolorowych i monochromatycznych). Ma na celu poprawę jakości obrazu, uwypuklenie cech obiektów będących celem późniejszej analizy.
3. Segmentacja, binaryzacja i przetwarzanie obrazu binarnego (filtracja, rozłączenie elementów stykających się, usunięcie obiektów przeciętych brzegiem)
4. Analiza obrazu - segmentacja oraz następnie wyznaczenie parametrów obiektów (np. rozkładów powierzchni ziaren, średnic Fereta)
5. Wizualizacja wyników.

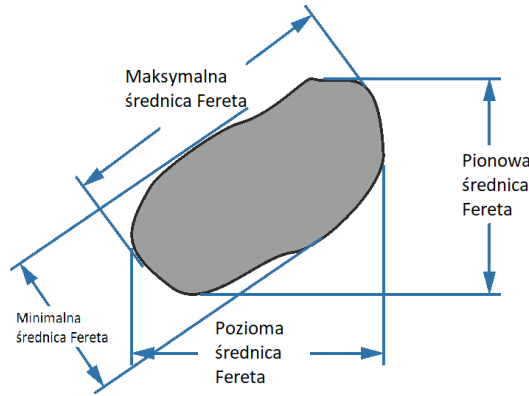
### 2 Parametry obiektów

Do najważniejszych parametrów obiektów zaliczamy :

- Liczebność obiektów (z uwzględnieniem efektu przecięcia brzegu). Jest to realizowane poprzez proces etykietowania.
- Liczba Eulera: liczba obiektów pomniejszona o ilość dziur w tych obiektach.
- Pola powierzchni, sumaryczne i rozkłady.
- Długości krawędzi (np. poprzez zliczanie punktów brzegowych, przybliżanie figury wielokątem, średniej z wewnętrznej i zewnętrznej krawędzi, formuły Croftona, ...)
- Długości rzutów: Rzutem figury  $D(\alpha)$  w kierunku wektora rzutowania  $\alpha$  nazywamy największą odległość pomiędzy wszystkimi prostymi, równoległymi do wektora  $\alpha$ , mającymi część wspólną z figurą  $D$ .
- Średnica zastępcza: średnica koła o polu powierzchni równym polu figury.
- Średnice Fereta: Długość najdłuższego oraz najkrótszego rzutu figury. Czasami wprowadza się też poziomą i pionową średnicę Fereta rozumianą jako długość rzutu poziomego i pionowego (Rys. 1).
- Wymiary fraktalne (np. kostkowy, korelacyjny, ...) - pozwalają na ocenę "poszarpania" obiektu.

---

<sup>1</sup>W większości na podstawie Tadeusiewicz, R., Korohoda, P. 1997 Komputerowa analiza i przetwarzanie obrazów



**Rysunek 1:** Średnice Fereta

- Momenty bezwładności n-tego rzędu dla pola powierzchni  $A(X)$  figury  $X$  :

$$M_{1X} = \frac{1}{A(X)} \sum_X x_i \quad (1)$$

$$M_{1Y} = \frac{1}{A(X)} \sum_X y_i \quad (2)$$

$$M_{2X} = \frac{1}{A(X)} \sum_X (x_i - M_{1X})^2 \quad (3)$$

$$M_{2Y} = \frac{1}{A(X)} \sum_X (y_i - M_{1Y})^2 \quad (4)$$

$$M_{2XY} = \frac{1}{A(X)} \sum_X (x_i - M_{1X})^2 \cdot (y_i - M_{1Y})^2 \quad (5)$$

- Współczynniki kształtu:

- Bezwymiarowy współczynnik kształtu ( $L$  - obwód,  $S$  - powierzchnia obiektu):

$$R_S = \frac{4\pi \cdot S}{L^2} \quad (6)$$

- Współczynnik Fereta  $R_F$  ( $F_{max}, F_{min}$  - maksymalna i minimalna średnica Fereta )

$$R_F = \frac{F_{min}}{F_{max}} \quad (7)$$

- Współczynnik cyrkularności:

$$R_{C1} = 2\sqrt{\frac{S}{\pi}}, \quad R_{C2} = \frac{L}{\pi} \quad (8)$$

- Współczynnik Blaira-Blissa

$$R_B = \frac{S}{\sqrt{2\pi \sum_i r_i^2}} \quad (9)$$

gdzie:

$S$  - pole powierzchni

$r_i$  - odległość  $i$ -tego piksela od środka ciężkości

## Dyskretne transformacje obrazu

### 1 Dyskretna Transformacja Cosinusowa (DCT)

Transformacja ta ma zastosowanie przede wszystkim do kompresji obrazów jpg oraz do konwersji mpeg. Para transformat cosinusowych dana jest wzorami:

$$B(p, q) = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N} \quad (1)$$

$$A(m, n) = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N} \quad (2)$$

gdzie:

$$0 \leq p \leq M-1, 0 \leq q \leq N-1$$

$$0 \leq m \leq M-1, 0 \leq n \leq N-1$$

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}} & \text{dla } p = 0 \\ \frac{2}{\sqrt{M}} & \text{dla } p \neq 0 \end{cases} \quad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}} & \text{dla } q = 0 \\ \frac{2}{\sqrt{N}} & \text{dla } q \neq 0 \end{cases}$$

Do implementacji DCT służy polecenie `dct2(obraz, m, n)`, a do transformacji odwrotnej `idct2(obraz, m, n)`. Jeżeli `[m,n] < size(obraz)`, obraz A jest przycinany.

#### 1.1 Kompresja JPEG '92

Kompresja JPEG '92 (ang. *Joint Photographic Experts Group*) oparta jest o transformację DCT2 wykonywaną w oknach o rozmiarze 8x8. Istotą procesu jest kwantyfikacja wyników transformaty oraz późniejsze kodowanie. Proces kompresji wygląda następująco:

1. Konwersja RGB do YCbCr połączona z przesunięciem średniej do 0.0:

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ Cr &= 128 - 0.168736 \cdot R - 0.331264 \cdot G + 0.53 \cdot B \\ Cb &= 128 + 0.5 \cdot R - 0.418688 \cdot G - 0.081312 \cdot B \end{aligned} \quad (3)$$

W MatLABie istnieje polecenie konwertujące `rgb2ycbcr()`. Wymagane jest tylko późniejsze, ręczne przesunięcie średniej.

2. Downsampling (redukcja danych)

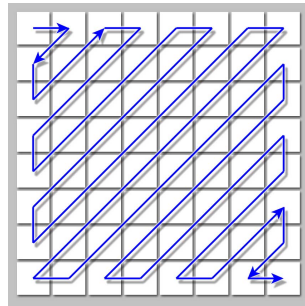
- 4:4:4 - Brak downsamplingu
- 4:2:2 - Redukcja poziomych składowych Cb i Cr o połowę
- 4:2:0 - Redukcja poziomych i pionowych składowych barwy o połowę.

3. Wykonywanie DCT2 w oknach 8x8.

4. Kwantyfikacja wyników przy wykorzystaniu tablic kodowania  $Q_Y$  i  $Q_C$ . Polega ona na podzieleniu każdego bloku przez tablicę i zaokrągleniu do najbliższej liczby całkowitej. Dopiero od tego punktu algorytm jest stratny.

$$Q_Y = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad Q_C = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 69 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

5. Zamiana tablicy na ciąg liczbowy. Wykorzystywana jest ścieżka zygzakowa, by jak najczęściej zer było koło siebie (fig.1).



**Rysunek 1:** Ścieżka stosowana w kompresji JPG. Źródło <http://en.wikipedia.org/wiki/Zigzag>

6. Kompresja. Stosowany jest najczęściej algorytm Huffmana, choć stosowane też może być kodowanie arytmetyczne (bardziej efektywna kompresja, ale kosztowniejsza obliczeniowo).

## 2 Transformacja Radona

Transformacja Radona podaje ilość pikseli obrazu binarnego o wartości logicznej prawda w rzucie na prostą umieszczoną pod kątem theta względem obrazu. Transformacja ta dana jest wzorem:

$$R_o(x') = \int_{-\infty}^{\infty} f(x' \cos \theta - y' \sin \theta, x' \sin \theta + y' \cos \theta) dy \quad (4)$$

gdzie:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Do transformacji Radona służy polecenie `[R,xp]=radon(obraz, theta)`, gdzie R - wynik transformacji Radona pod kątem theta (kiedy theta - wektor, R - macierz) xp - wektor zawierający współrzędne kątowe odnoszące się do macierzy R

Do transformacji odwrotnej służy polecenie `iradon(R, theta, parametry)`. Do parametrów zaliczamy:

- metoda interpolacji: 'nearest', 'linear', 'spline'
- metoda filtracji: 'Ram-Lak', 'Shepp-Logan', 'Cosine', 'Hamming', 'Hann'
- współczynnik skali osi częstotliwości: 0-1, domyślnie 1;
- rozmiar wyjściowy: domyślnie:  $n = 2 \cdot \text{floor}(\text{size}(R,1)/(2 * \sqrt{2}))$

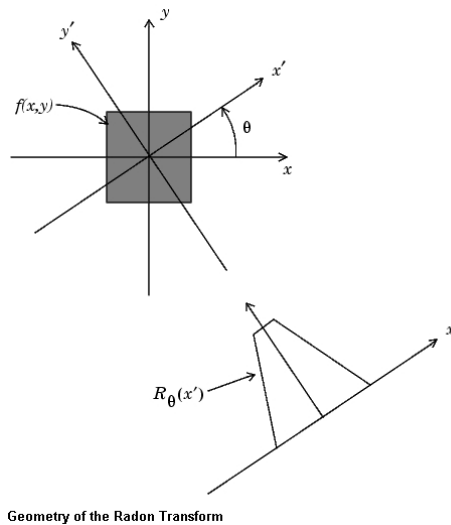
Odwrotna Transformata Radona, w wersji monochromatycznej, jest stosowana w tomografii medycznej do inwersji danych pomiarowych.

## 3 Transformacja Hougha

Standardowa Transformacja Hougha (w pakiecie MatLAB dostępna jest tylko dla obrazów logicznych), wykorzystuje parametryczną postać prostej:

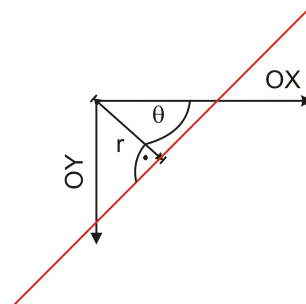
$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (5)$$





**Rysunek 2:** Idea Transformacji Radona(Źródło: MATLAB help: Radon transform::description of)

Na fig.3 przedstawiono interpretację geometryczną postaci parametrycznej prostej. Wartość  $r$  jest najkrótszą odległością dowolnego punktu prostej od środka układu współrzędnych (czyli długością linii prostopadłej do prostej przechodzącej przez punkt  $(0,0)$ ), a kąt  $\theta$  jest kątem pomiędzy dodatnią półosią  $OX$ , a linią  $r$ .



**Rysunek 3:** Interpretacja geometryczna parametrycznej postaci prostej

Dzięki temu możemy każdemu punktowi przypisać pewną funkcję sinusoidalną. Linia jest reprezentowana jako punkt przecięcia sinusoid pochodzących od punktów leżących na owej linii. Do implementacji Transformacji Hougha służy polecenie: `[H, theta, rho]=hough(BW, parametry)`. Do parametrów zaliczamy:

- 'ThetaResolution' - Liczba rzeczywista z przedziału 0-90, określająca odległość pomiędzy kolejnymi kątami  $\theta$ , domyślnie 1;
- 'RhoResolution' - liczba z przedziału: 0 –  $norm(size(BW))$ . Domyślnie 1.

Funkcjami powiązanymi z transformacją Hougha są:

- `peaks = houghpeaks(H, numpeaks, parametry)` - służy ona do lokalizacji punktów przecięć sinusoid na obrazie;
- `houghlines(BW, theta, rho, peaks)` - służy do detekcji linii prostych na obrazie logicznym.

## Dwuwymiarowa Transformata Fouriera

### 1 Dwuwymiarowa Szybka Transformata Fouriera

Dwuwymiarowa transformata Fouriera (FFT2) służy przejściu z domeny przestrzeni do domeny częstości. W tym przypadku częstotliwość (częstość) możemy definiować w formie dwóch częstotliwości ortogonalnych (częstotliwość pozioma i pionowa) lub jako częstotliwość wypadkową rozumianą jako odległość od środka obrazu transformaty (po przesunięciu widma ku środkowi).

Dwuwymiarowa Szybka Transformata Fouriera obrazu **I** dana jest wzorem:

$$F(nz, nx) = \beta_L \cdot \sum_{kz=0}^{Nz-1} \left[ \sum_{kx=0}^{Nx-1} I(kz, kx) \cdot \exp\left(\frac{-2\pi i \cdot kx \cdot nx}{Nx}\right) \right] \cdot \exp\left(\frac{-2\pi i \cdot kz \cdot nz}{Nz}\right) \quad (1)$$

Odwrotna transformata Fouriera dla  $F_{Obrazu}$  **F** jest dana wzorem:

$$I(kz, kx) = \beta_F \cdot \sum_{nz=0}^{Nz-1} \left[ \sum_{nx=0}^{Nx-1} F(nz, nx) \cdot \exp\left(\frac{2\pi i \cdot nx \cdot kx}{Nx}\right) \right] \cdot \exp\left(\frac{2\pi i \cdot nz \cdot kz}{Nz}\right) \quad (2)$$

gdzie  $\beta_L \cdot \beta_F = \frac{1}{M \cdot N}$ . Najczęściej  $\beta_L = 1$  i  $\beta_F = \frac{1}{MN}$ .  $i = \sqrt{-1}$ .

Transformacje te wykonywane są poleceniem: **F=fft2(I)** oraz **I=ifft2(F)** odpowiednio dla prostej i odwrotnej transformaty.

FFT2 wykonywana jest dwuetapowo. Najpierw dokonywany jest szereg transformat jednowymiarowych (np. w wierszach), a następnie dla ortogonalnego wymiaru wykonywane są transformaty na wynikach otrzymanych w poprzednim kroku. Z tego powodu poniższe polecenia MatLABa są równoważne:

**fft2(I)==fft(fft(I).').';**

### 2 Widma FFT2

Dla dwuwymiarowej FFT, podobnie jak dla FFT 1D możemy policzyć widma amplitudowe (WA) i fazowe (WF). Dane są one wzorami:

$$WA(m, n) = \sqrt{Im(F(m, n))^2 + Re(F(m, n))^2} \quad (3)$$

$$WF(m, n) = \arctg\left(\frac{Im(F(m, n))}{Re(F(m, n))}\right) \quad (4)$$

Do policzenia ich służą polecenia MatLABa:

**WA=abs(F)** oraz **WF=angle(F)**.

Ponieważ energia **F** skupiona jest na brzegach (składowa zero-częstotliwościowa), niezbędne jest przesunięcie jej do centrum. Służy do tego para poleceń: **FS=fftshift(F)** oraz **F=ifftshift(FS)**.

Uwaga! Polecenia te są wykonywane tylko i wyłącznie na wynikach transformaty oraz nie są symetryczne, tzn. poniższe polecenia nie są równoważne:

**fftshift(fftshift(F))  $\neq$  ifftshift(fftshift(F))**

### 3 Filtracja w domenie częstotliwości

Głównym zastosowaniem FFT2 jest filtracja obrazów. Polega to na zamianie kosztownej obliczeniowo operacji splotu na znacznie bardziej szybkie mnożenie widm. Podobnie jak w przypadku filtracji sygnałów jednowymiarowych, możemy podzielić filtry na:

- dolnoprzepustowe - przepuszczające dolne częstotliwości, reprezentowane na ogół poprzez trend na obrazie (np. stopniowe zmiany jasności tła);
- górnoprzepustowe - przepuszczające wysokie częstotliwości, reprezentujące na ogół krawędzie, szum i inne krótkookresowe zmiany jasności na obrazie;
- pasmoprzepustowe i pasmowycięciowe - przepuszczające lub wycinające pewien przedział częstotliwości;
- wycięciowe - eliminujące pojedynczą częstotliwość z obrazu (np. efekt pochodzący od sieci elektrycznej).

Filtracja obrazu  $I$  przy użyciu filtra  $M$  dana jest wzorem:

$$FF = ifft2(|WA(I) \cdot WA(M)| \cdot e^{i(WF(I)+WF(M))}) \quad (5)$$

Wynika z tego rozmiar maski filtru musi być taki sam jak obrazu. Istnieją dwie metody projektowania filtru: bezpośrednio w 2D oraz poprzez rozszerzenie filtrów 1D w 2D.

#### 3.1 Projektowanie filtrów 2D

Idealny, zerofazowy filtr dolnoprzepustowy ma postać centralnie położonego okręgu (o wartościach 1), gdzie długość promienia koła określa pasmo przepuszczania.

Innym przykładem filtru 2D jest dolnoprzepustowy filtr Butterwortha opisany wzorem:

$$H(z, x) = \frac{1}{1 + (D(z, x)/D_0)^{2n}} \quad (6)$$

$$D(z, x) = \sqrt{(x - Nx/2)^2 + (z - Nz/2)^2}, \quad (7)$$

gdzie  $D_0$  odpowiada za częstotliwość odcięcia,  $n$  jest rzędem filtru i odpowiada za krzywiznę zbocza filtru, a  $[Nz, Nx]$  - rozmiar obrazu.

Filtr Butterwortha w wariancie pasmowozaporowym o szerokości  $W$  dany jest wzorem:

$$H(z, x) = \frac{1}{1 + \left( \frac{D(z, x) \cdot W}{D(z, x)^2 - D_0^2} \right)^{2n}} \quad (8)$$

Kolejnym przykładem filtru dolnoprzepustowego jest filtr Gaussowski, dany wzorem:

$$H(z, x) = \exp\left(\frac{-D^2(z, x)}{2 \cdot D_0^2}\right) \quad (9)$$

Do zmiany filtrów 1D w filtry 2D służą następujące polecenia:

- `filtr_1D=remez(N, [częstotliwości], [amplitudy])`, gdzie  $N$  jest liczbą współczynników (musi być parzysta). Jako częstotliwości podaje się wartości z przedziału 0-1.
- `filtr_2D=ftrans2(filtr_1D)` - stworzenie osiowo symetrycznego filtru 2D na podstawie filtru 1D.
- `filter2(filtr_2D, obraz)` - filtracja obrazu.

## 4 Korelacja

Jednym z zastosowań FFT jest funkcja korelacji służąca do wyszukiwania elementów podobnych na obrazie. Do liczenia współczynnika korelacji służy polecenie:

`corr2(obraz1, obraz2);`.

Współczynnik korelacji jest liczony wg następującego wzoru:

$$r = \frac{\sum_n \sum_m (A_{mn} - \bar{A}) \cdot (B_{mn} - \bar{B})}{\sqrt{(\sum_n \sum_m (A_{mn} - \bar{A})^2) \cdot (\sum_n \sum_m (B_{mn} - \bar{B})^2)}} \quad (10)$$

$\bar{A}$  - wartość średnia obrazu A.

W praktyce sprowadza się to do wyszukiwania maksimum na wyniku polecenie:

`real(ifft2(fft2(obraz).*fft2(rot90(worzec, 2), size(obraz))))`

## 5 Transformata Falkowa 2D

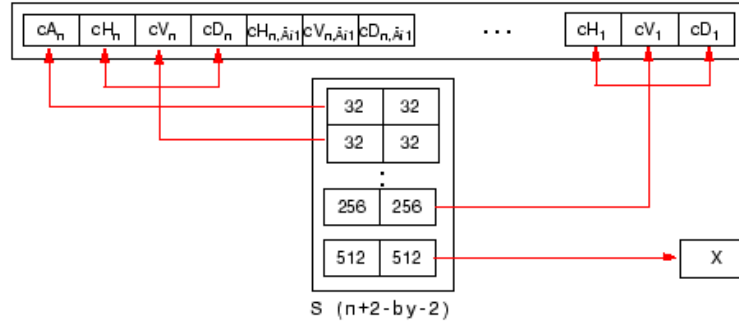
Transformata falkowa jest przykładem analizy czasowo-częstotliwościowej sygnału. Polega na dekompozycji sygnału na część aproksymującą i detale przy użyciu konkretnej falki (wavelet). Odwrotna transformata falkowa polega na rekonstrukcji sygnału bazując na jego składowych. Podobnie jak transformata Fouriera, występuje ona w formie ciągłej (cwt) i dyskretniej (dwt). Transformata falkowa ma zastosowanie w kompresji (np. JPEG 2000), odsumowaniu sygnałów czy fuzji obrazów. W przypadku obrazów monochromatycznych dekompozycja falkowa odbywa się z wykorzystaniem funkcji: `[A,H,V,D]=dwt2(X,falka)` dla 1 rzędu i `[C,S]=wavedec2(x, poziom, falka)` dla dowolnego rzędu. W wyniku dekompozycji otrzymujemy 4 obrazy: (A - aproksymacja, H - detale poziome, V - detale pionowe, D - detale ukośne). Rekonstrukcja odbywa się z wykorzystaniem funkcji: `idwt2` oraz `waverec2`.

A3	H3	H2	H1
V3	D3		
V2		D2	
V1			D1

**Rys. 1:** Idea dekompozycji falkowej obrazu na 3 poziomie.

Podobnie jak w przypadku sygnałów 1D, tu również dekompozycji na wyższych poziomach podlega tylko i wyłącznie część aproksymacyjna (rys. 1). W wyniku jej działania otrzymujemy submacierze o wymiarze dwukrotnie mniejszym, które są zapisane w wektorze **C** (rys. 2). Do ekstrakcji danych z wektora **C** warto skorzystać z funkcji: `A=appcoef2(C, S, falka, poziom)` do uzyskania macierzy aproksymacji

i funkcji `detcoef2(opcja, C, S, poziom)` do uzyskania macierzy detali na określonym poziomie. W tym drugim przypadku opcje to 'all', jeśli chcemy uzyskać wszystkie 3 macierze detali lub 'h', 'v', 'd' dla pojedynczych macierzy.



**Rys. 2:** Struktura danych wyjściowych w funkcji `wavedec2`. Źródło: MatLAB Wavelet Toolbox.

## 6 Transformata Gabora

Inną transformatą czasowo-częstotliwościową jest transformata Gabora. Ma ona zastosowanie głównie do detekcji tekstur na obrazach. Polega ona na splocie zbioru kierunkowych masek (tzw. bank filtrów) z obrazem. Maska filtru ma postać zespoloną i stanowi złożenie dwóch elementów: sinusoidalnej fali nośnej 2D o określonej długości fali  $\lambda$  i orientacji  $\theta$  oraz dwuwymiarowej funkcji Gaussa o zadanym odchyleniu  $\sigma$ , współczynnika eliptyczności  $\gamma$ .

$$G(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 \cdot y'^2}{2\sigma^2}\right) \cdot \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \quad (11)$$

gdzie

$$\begin{cases} x' = x \cdot \cos \theta + y \cdot \sin \theta \\ y' = -x \cdot \sin \theta + y \cdot \cos \theta \end{cases} \quad (12)$$

W programie MatLAB transformację Gabora wykonuje się w dwóch krokach:

1. Tworzymy bank filtrów `g` korzystając z polecenia:  
`g = gabor(długość fali, kąty, 'SpatialFrequencyBand', 1.0, 'SpatialAspectRatio', 0.5);`  
Długość fali podaje w pikselach/okres, natomiast kąty (orientacje filtrów) w stopniach. Ilość filtrów w banku jest iloczynem ilości długości fal i ilości orientacji. Dwa ostatnie parametry są opcjonalne i podane z wartościami domyślnymi. Odpowiadają one za częstotliwość fali nośnej (sinusoidalnej) i za rozciągnięcie funkcji Gaussa (stosunek długości osi prostopadłej do kąta orientacji do długości osi zgodnej z kątem orientacji).
2. Filtrując obraz korzystając z polecenia:  
`Magn = imgaborfilt(obraz, bank filtrów);`  
W wyniku filtracji dostajemy zbiór obrazów odpowiadający każdemu z filtrów dostępnemu w banku (najpierw dla pierwszej orientacji wszystkie długości fal, potem dla każdej kolejnej orientacji odpowiadające im długości).