

## 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 `[mapa, legenda]=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:

- obraz logiczny, przyjmujący wartości `true` / `false`;
- obraz monochromatyczny - tablica poziomów szarości. Wartości mogą być typu `double` (0-1), `uint8` (0-255) lub `uint16`;
- obraz kolorowy RGB - potrójna tablica intensywności (3 tablice monochromatyczne);
- obraz kolorowy indeksowany - składa się z dwóch części: mapy i legendy, gdzie legenda jest tablicą o wymiarze  $N \times 3$  (gdzie  $N$  - ilość dostępnych kolorów), a mapa jest tablicą z numerami kolorów (numerami odpowiednich wersów w legendzie).
- obrazy sekwencyjne (`ruhome`) - tablica 4D, gdzie czwarty wymiar odpowiada za numer klatki (obrazu w sekwencji).

### 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)
- HSV (Hue, Saturation, Value) - oparte o stożek:  $H=0:360$ ;  $S,V=0:100$ .
- CMY

- $L^*a^*b^*$  - (Luminance, a i b)
  - HLS (Hue, Saturation, Lightness)
3. Poczwórna tablica: CMYK (Cyan, Magenta, Yellow, Key (black))
  4. n-kanałowe.

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.easyrgb.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

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

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

$$\text{CMY} = 255 - \text{RGB} \text{ (dla uint8)} \text{ i } \text{RGB} = 1 - \text{CMY} \text{ (dla double)}.$$

## 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)`. Wybrana paleta stosowana jest do wszystkich obrazów logicznych i monochromatycznych 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 tożsamy 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 & Koproński):

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

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`).