

[upel.agh.edu.pl](https://upel.agh.edu.pl)

# SW: Instrukcja - Rozdzielczość obrazów

6 — 8 minut

---

## Instrukcja - Rozdzielczość obrazów

Autorzy: Tomasz Kryjak, Tomasz Pięciak, Piotr Pawlik

Tematyka:

- zapoznanie z pojęciem rozdzielczości przestrzennej (rozmiaru),
- zapoznanie z pojęciem rozdzielczości dpi (**dots per inch**),
- zapoznanie z pojęciem rozdzielczości poziomów jasności (dla obrazów w skali szarości),
- zapoznanie się z obracaniem i transformacjami przestrzennymi obrazów

## Ćwiczenie:

### A. Rozdzielczość przestrzenna - powiększanie obrazu

Rozdzielczość przestrzenna to ilość pikseli z których składa się obraz (inaczej gęstość punktów próbkowania). Rozdzielczość obrazu można modyfikować (zmniejszać/zwiększać) co często nazywamy skalowaniem obrazu. Przeskalowanie do innego

rozmiaru nie zwiększa ilości informacji, zwiększa się tylko ilość pikseli - cała operacja powoduje zniekształcenia.

Aby wyznaczyć wartości "brakujących" pikseli wykorzystuje się metody:

- najbliższego sąsiada (*nearest neighbor*),
- interpolacji (dwuliniowej (*bilinear*), dwukubicznej (*bicubic*) )

### Ciekawostki:

- omawiana rozdzielczość przestrzenna w programach graficznych określana jest jako **Rozmiar Obrazka (Image Size)** (Gimp i Corel Photo-Paint) lub **Pixel Dimension** (Photoshop). Pod pojęciem rozdzielczość (**Resolution**) rozumie się parametr dpi (**dots per inch**) - o którym w dalszej części ćwiczenia.
  - w programach Gimp i Photoshop istnieje możliwość wyboru metody zmiany rozdzielczości (interpolacja lub najbliższy sąsiad).
1. Otwórz program **Matlab**. Ustal ścieżkę **Current Directory** na swój własny katalog. Utwórz nowy m-plik i umieść w nim polecenia `close all;clear all;`
  2. Wczytaj obraz "lena100.bmp". Do zmiany rozdzielczości przestrzennej służy funkcja `imresize`. Zapoznaj się z dokumentacją do niej. Zauważ że istnieje możliwość wybrania metody interpolacji (*Interpolation method*) oraz określenia kształtu branego pod uwagę otoczenia (*Interpolation kernel*).
  3. Przeskaluj obraz *lena100* do rozmiaru 200x200 pikseli, wykorzystując 3 metody: najbliższego sąsiada, dwuliniową, dwukubiczną. Wyniki wyświetl na osobnych "figurach" - umieszczenie wszystkiego na jednym obrazie poprzez subplot spowoduje zmianę rozmiaru wyświetlanych obrazków.

4. Skutki zmiany rozmiaru inaczej prezentują się na obrazach sztucznych. Wczytaj obraz "szachownica.bmp". Ma on rozmiar 16x16 pikseli. Przeskaluj (trzema metodami) do rozmiaru 160x160, a później obraz 160x160 do 100x100. Zaobserwuj działanie poszczególnych metod interpolacji. Która z metod interpolacji dała najlepszy wynik?
5. Zaprezentuj wyniki prowadzącemu.

### **B. Rozdzielczość przestrzenna - pomniejszanie obrazu**

1. Wczytaj obraz "lena.bmp". Ma on rozmiar 512x512. Wykorzystując funkcję `imresize` stwórz obrazy o rozmiarach 256x256, 128x128, 64x64 - metoda interpolacji jest w tym wypadku mniej istotna.
2. Wyświetlając obrazy "wymusimy" zachowanie rozmiaru na ekranie 512x512. W tym celu wykorzystamy parametr funkcji `imshow` `'InitialMagnification'`, ustawiamy ją odpowiednio na 200, 400, 800. Zapoznaj się z uzyskanymi wynikami. (wyświetl obrazy oryginalny i po przeskalowaniach)
3. Zaprezentuj wyniki prowadzącemu.

### **C. Ilość poziomów jasności**

Dla obrazów w skali szarości pojedynczy piksel zapisuje się zazwyczaj na 8 bitach, co daje 256 rozróżnialnych poziomów szarości. Dla większości zastosowań wartość ta jest wystarczająca. Oko ludzkie nie potrafi rozróżnić wszystkich 256 poziomów jasności (jest za mało czułe). Zazwyczaj człowiek rozróżnia 20-30 poziomów szarości.

1. Wczytaj obraz "lena.bmp". Zmniejsz jego rozmiar do 128x128 (łatwiejsze wyświetlanie). Wykorzystując funkcję `imadjust` w wersji:

- `imadjust(l,[low_in;high_in],[low_out; high_out])`

lub po prostu dzieląc każdy piksel przez liczbę uśrednianych poziomów

zmień ilość poziomów szarości z 0 - 255 na:

- 0-31
- 0-15
- 0-7
- 0-3
- 0-1 (binaryzacja)

Uwaga: funkcja `imadjust` jako parametr przyjmuje wartości z zakresu 0-1. Należy dokonać odpowiedniego przeskalowania.

2. Rezultaty wyświetl na wspólnym rysunku (funkcja `subplot`). Uwaga: Aby poprawnie wyświetlić obrazek należy wykorzystać następującą postać funkcji `imshow(lena,[])`; Czy rezultaty eksperymentu pasują do teorii o rozpoznawaniu przez człowieka ograniczonego zakresu poziomów jasności? Wizualne porównanie których obrazów o tym świadczy?
3. Zaprezentuj wyniki prowadzącemu.

## **D. Obracanie i transformacje przestrzenne obrazów**

1. Dla obrazu "lena.bmp" przetestuj działanie funkcji `imrotate`, obracając ten obraz o kąty: +90, +45, -90, -45 (po zapoznaniu się z dokumentacją)
2. Obraz "szachownica.bmp" obróć o -45 stopni, sprawdź i wyświetl wyniki dla 3 metod interpolacji: najbliższego sąsiada, dwuliniowej i

dwukubicznej. Która z metod interpolacji dała najlepszy wynik?

Odpowiedź wpisz do m-pliku w komentarzu zaczynającym się od:

% ODPOWIEDŹ:

3. Zapoznaj się z dokumentacją funkcji `imwarp`. Wykorzystamy najprostszą postać: `B=imwarp(A,tform)`; Do stworzenia struktury `TFORM` wykorzystaj klasę `affine2d` - zapoznaj się z jej dokumentacją. Dokonaj przekształcenia obrazu "lena.bmp" polegającego na dwukrotnym pomniejszeniu i jednoczesnym obrocie o -45 stopni (obie operacje w jednym wywołaniu `imwarp`). Użyj odpowiednich macierzy transformacji.
4. Użyj funkcji `imtranslate` do przesunięcia obrazu 'lena.bmp' o wektor [10, 25]. Spróbuj uzyskać taki sam efekt używając funkcji `imwarp`.
  - UWAGA - funkcja `imwarp` "przesuwa" współrzędne okno wyświetlania obrazu tak, aby wynik był całkowicie widoczny. W wypadku translacji oznacza to brak zmian w wyświetlanym obrazie - okno "przesunie" się o tyle o ile przesuniemy obraz. Dlatego przy translacji trzeba ustawić okno wyświetlania w pierwotnym położeniu za pomocą parametru 'OutputView':
    - `imwarp(A, tform, 'OutputView', imref2d(size(A)))`;
5. Zaprezentuj wyniki prowadzącemu.

Ostatnia modyfikacja: wtorek, 6 marzec 2018, 10:18