

1 — Laboratorium wprowadzające

1.1 Tematyka

- zapoznanie ze środowiskiem programistycznym wykorzystywanym podczas ćwiczeń – Matlab oraz pakiet Image Processing Toolbox,
- podstawy pisania skryptów w Matlab'ie (przypomnienie),
- wykorzystanie pomocy w pakiecie Matlab,
- operacje wejścia/wyjścia na obrazach (wczytywanie, wyświetlanie, zapisywanie),
- obrazy indeksowane.

1.2 Operacje wejścia/wyjścia

1. Uruchom program Matlab.
2. Uruchom pomoc programu Matlab (Help). Wybierz *Image Processing Toolbox*, a następnie *Import, Export and Conversion*. Podczas laboratoriów pomoc będzie często wykorzystywana. Uwaga. Pomoc do pakietu Matlab jest również dostępna na stronie: <http://www.mathworks.com/help/matlab/index.html>.
3. Odszukaj temat *Basic Import and Export*.
4. Zapoznaj się ze sposobem uzyskiwania informacji o pliku graficznym (`imfinfo` oraz *Getting Information About a Graphics File*).
5. Ustal ścieżkę *Current Folder* na swój własny katalog. Okno znajduje się domyślnie po lewej stronie. Utwórz nowy m-plik (*New->Script*). Wykorzystując informacje o funkcji `imfinfo` odczytaj parametry plików *lena.bmp* oraz *lena.jpg* (pliki znajdują się w dostępnym na platformie kursu archiwum, które należy rozpakować w „swoim” katalogu):
 - zapisz utworzony m-plik (w swoim katalogu). Zapisanie możliwe jest dopiero po uprzedniej edycji pliku (np. napisaniu jednej literki). Proszę zwrócić uwagę na poprawną nazwę – **nie powinna** zaczynać się od liczby i być słowem kluczowym lub nazwą funkcji wbudowanej, a także na poprawne rozszerzenie.
 - przypomnienie: komentarze w Matlab'ie to znak `%`. Ciekawą opcją jest tworzenie sekcji kodu (*section*). Sekcja zaczyna się od znaków `%%` np. `%% Wczytanie` (w edytorze obszar sekcji podświetla się). Sekcja kończy się rozpoczęciem następnej sekcji. Sekcje umożliwiają podzielenie kodu (skryptu) na fragmenty oraz wykonywanie poszczególnych sekcji. (*Editor->Run Section* lub `Ctrl+Enter`). Zaleca się korzystanie z tego mechanizmu, ponieważ pozwala on na porządkowanie tworzonego kodu,
 - kolejną ważną opcją jest możliwość wykonania zaznaczonych fragmentów kodu (*Evaluate Selection* – klawisz F9),
 - wpisz polecenie `imfinfo` z odpowiednimi parametrami (przypomnienie: jeżeli po funkcji nie umieści się średnika, jej wynik wyświetli się w konsoli),
 - odpowiedź: jeśli ustawi się kursor na poleceniu i naciśnie F1 to wyświetli się pomoc do funkcji,
 - zapisz m-plik,
 - uruchom go (*Run* lub F5),
 - przeanalizuj różnice w parametrach obu plików.

6

Laboratorium wprowadzające

6. Zapoznaj się ze sposobem wczytywania (`imread` – *Reading Image Data*) i wyświetlania obrazów (`imshow` – *Display and Exploration/Basic Display*). Wczytaj pliki *lena.bmp* i *lena.jpg* oraz wyświetl je. Przy wielu obrazkach/wykresach bardzo pomocne jest polecenie `figure(n)`, gdzie `n` – numer wykresu.
 - zwróć uwagę na zakładkę *Workspace* – pokazane są tam wszystkie zmienne aktualnie załadowane do przestrzeni roboczej,
 - zwróć uwagę na format danych w jakim jest zapisane są piksele (`uint8`),
 - pomocne są też polecenia: `clearvars`, `close all`, `clc` – pierwsze czyści zmienne wczytane do przestrzeni roboczej, drugie zamyka wszystkie okienka (wykresy), a trzecie czyści konsolę. Zazwyczaj dobrze jest je umieścić na początku m-pliku (uwaga: polecenie `clear all` czyści też „brakepointy”, dlatego zwykle lepiej jest stosować `clearvars`).
7. W praktycznych rozważaniach często analizuje się obrazy w odcieniach szarości (ang. *grayscale*).
 - dokonaj konwersji obrazu *lena.bmp* do skali szarości – funkcja `rgb2gray` (sprawdź w pomocy),
 - wyświetl uzyskany obraz (na kolejnym wykresie – np. nr 3).
8. Czasami konieczne jest zapisanie przetworzonego obrazu. Służy do tego funkcja `imwrite`. Zapisz obraz *lena* w skali szarości, w formacie 'bmp'. Zwróć uwagę, aby nie nadpisać oryginalnego obrazu kolorowego (RGB) – zastosuj nazwę *lena_gray.bmp*.
9. Obraz w skali szarości możemy traktować jako dwuwymiarową funkcję $L(x,y)$, gdzie x, y to współrzędne piksela, a $L(x,y)$ poziom jasności (najczęściej [0;255] – zapis na 8-bitach, typ *unsigned int*). Wyświetl obraz *lena.bmp* jako funkcję dwuwymiarową:
 - przydatne jest polecenie `mesh`,
 - przed poleceniem `mesh` można (warto) ustawić mapę kolorów na odcienie szarości poleceniem: `colormap gray`,
 - obraz wyświetl na wykresie nr 4,
 - na wykresie użyteczna jest opcja rotacji 3D (*Tools->Rotate 3D*).
 - Podczas przetwarzania i analizy obrazów przydatne bywają "przekroje" przez obraz, czyli wartości funkcji $L(x,y)$ w przypadku gdy x lub y jest ustalone. Wykonaj jeden wybrany przekrój w x i y :
 - wykorzystaj polecenie: `plot(lena(10,:))`;
 - oznacza ono wybór 10 wiersza i całej kolumny – znak ':';
 - analogicznie wykonaj przekrój „wzdłuż” wybranej kolumny,
 - przekroje wyświetl na wykresach 5 i 6,
 - warto w tym miejscu zwrócić uwagę, że w Matlab'ie pierwsza współrzędna to numer wiersza (tj. położenie w pionie), a druga to numer kolumny (tj. położenie w poziomie). Jest to niejako sprzeczne z powszechnie obowiązującym podawaniem parametru rozdzielczości, gdzie zwykle pierwsza liczba to liczba kolumn, a druga to liczba wierszy. Przykładowo dla rozdzielczości HD 1920 × 1080 – kolumny × wiersze – poziom × pion.

Obrazy indeksowane (ang. *indexed images*), inaczej pseudokolorowe

Obrazy indeksowane składają się z dwóch macierzy – obrazu oraz tzw. mapy kolorów. W macierzy obrazu zapisane są wartości poszczególnych pikseli. Macierz mapy kolorów ma rozmiar $m \times 3$ (m wierszy, w każdym 3 wartości – składowe R,G,B). Podczas wyświetlania, na podstawie wartości piksela, odczytywany jest kolor z macierzy mapy kolorów. W ten sposób możliwe staje się "pokolorowanie" obrazu w skali szarości. Wykorzystanie pseudokoloru nie wpływa na ilość informacji zawartej na obrazku, pomaga jedynie przedstawić go w bardziej czytelnej (dla

PRZF

1.3 Obrazy indeksowane (ang. *indexed images*), inaczej pseudokolorowe 7

człowieka) formie. Szczegółowy opis w pomocy Matlab'a (*Image Processing Toolbox->Import, Export and Conversion->Image Type Conversion*).

1. Utwórz nowy m-plik.
2. Wczytaj obraz *lena_gray.bmp* oraz przekonwertuj go do obrazu indeksowanego. Wykorzystaj funkcję `gray2ind`. Rozmiar mapy kolorów określ na 256. Jako wynik działania zwrócony zostanie obraz oraz mapa kolorów.
3. Zobacz jaką postać ma mapa kolorów – np. wyświetl w oknie poleceń (wpisując nazwę zmiennej w której zapisana jest mapa) lub klikając na odpowiednią zmienną w oknie *Workspace*. Przypatrz się wyświetlonym wartościom. Zauważ, że wartości się powtarzają.
4. Wyświetl obraz indeksowany. Wykorzystaj funkcję `imshow(X,map)` w wersji z podaniem mapy kolorów. Obraz indeksowany nie powinien różnić się od wersji w odcieniach szarości.
5. Zmień mapę kolorów. Wykorzystaj polecenie `map = colormap(...)`. W pakiecie *Image Processing* dostępne są predefiniowane mapy kolorów (pomoc do funkcji `colormap`). Zadbaj o to aby mapa składała się z 256 wierszy. Wyświetl obraz z nową mapą kolorów.
6. Warto wiedzieć, że można tworzyć własne mapy kolorów – w Matlabie dostępne jest do tego specjalne narzędzie (`colormapeditor`).