Tomasz Kryjak

Podstawowe operacje przeprowadzane na obrazach cyfrowych. Przekształcenia punktowe.

Cel:

- zapoznanie z podstawowymi operacjami przeprowadzanymi na obrazach cyfrowych:
  - typu LUT (operacja jednoargumentowa)
  - arytmetycznymi (operacje dwuargumentowe): dodawanie, odejmowanie, mnożenie, dzielenie
  - logicznymi (operacje jedno i dwuargumentowe): AND, OR, XOR, NOT

Operacja LUT - polega na przekształcaniu wartości poszczególnych pikseli obrazu przy użyciu z góry przygotowanych tabel przekodowań (tabel korekcji).

1. Otwórz program Matlab. Ustal ścieżkę Current Directory na swój własny katalog na dysku D. Utwórz nowy m-plik (**New->Script**) lub (**New Script**). Porady:
   - aby uniknąć 10 otwartych okienek z obrazkami (Figure) dobrze jest każdą operację wykonywać w osobnym m-pliku.
   - przed właściwym kodem w m-pliku dobrze jest umieścić polecenia `clearvars; close all; clc;` Spowodują one wyczyszczenie przestrzeni roboczej Matlaba (Workspace), zamknięcie wszystkich okienek typu *Figure* oraz wyczyszczenie konsoli.
   - porady dotyczą tego i przyszłych ćwiczeń

   Zapisz i nazwij utworzony m-plik. (Zapis możliwy jest dopiero po dodaniu jakieś zawartości).

2. Operację LUT realizuje się za pomocą funkcji `intlut`. Przy czym najważniejsze jest stworzenie odpowiedniej tablicy przekodowania. W przetwarzaniu obrazów najczęściej wykorzystuje się następujące funkcje:
   - typu kwadratowa, pierwiastek kwadratowy
   - typu logarytm, odwrócony logarytm
   - typu wykładnicza,
   - inne (np. piłokształtna).

3. Wczytaj przygotowany plik z przekodowaniami LUT. (wcześniej ściągnij archiwum ze strony www i rozpakuj w odpowiednim katalogu). Wykorzystaj polecenie: `load funkcjeLUT;` . Przekodowań jest siedem. Zostaną one wczytane do przestrzeni roboczej. Ich nazwy widoczne są w okienku **Workspace.**

4. Wyświetl przykładową funkcję. Wykorzystaj polecenie `plot` (np. `plot(kwadratowa);`) (Wcześniej dobrze jest zadeklarować numer wykresu - np. `figure(1);`)

5. Wczytaj przykładowy obraz i wyświetl go - do wyboru "lena.bmp" lub "jet.bmp". Przypomnienie: wczytywanie `imread`, wyświetlanie `imshow`, kolejna figure - `figure(numer)`.

6. Na wybranym obrazie wykonaj operację LUT - na początek z tablicą przekodowań "kwadratowa". Wynik wyświetl. Podpowiedź: sprawdź w helpie jak działa funkcja `intlut`.

7. Aby lepiej zobaczyć w jaki sposób działają różne przekodowania LUT skonstruujemy funkcję, która jako argumenty pobierać będzie obrazek oryginalny oraz tablicę przekodowania, a następnie na wspólnym rysunku będzie wyświetlać: funkcję, obraz wejściowy oraz wynik przekodowania. (Przy okazji zobaczymy/przypomnimy sobie jak tworzy się funkcje w Matlabie oraz poznamy/przypomnimy/utrwalimy polecenie `subplot`).
   - utwórz nową m-funkcję (**New->Function**),
   - pole z argumentami wyjściowymi [`output_args`] proszę usunąć (znak '=" też),
   - nadaj funkcji nazwę: `LUT`,
   - jako argumenty wyjściowe wybierz (`obraz, przekodowanie`),
   - zapisz m-plik, Matlab sam zaproponuje nazwę LUT.m,
   - wykonaj przekodowanie LUT - dokładnie tak jak w punkcie 6,
   - wyświetl wyniki:
     - subplot powinien składać się z trzech pól - wykres przekodowania i dwa obrazy (oryginalny i przekształcony),
     - można zastosować układ 2x2 i górny wykres połączyć (`subplot(2,2,1:2)`) lub układ 1 x 3,
     - każdy wykres powinien być podpisany (`title`),
     - aby wykres przekodowania wyglądał "porządnie" można wykorzystać następujące funkcje: `xlim, ylim, daspect`. Szczegóły w helpie.

   - oczywiście kod należy umieścić pomiędzy nagłówkiem funkcji a słowem `end`.

8. W "głównym" m-pliku wywołaj stworzoną funkcję. Najpierw utwórz *figure* z odpowiednim numerem a następnie wywołaj funkcję LUT z odpowiednimi argumentami. Aby przejrzeć wyniki wszystkich przekodowań konieczne jest stworzenie siedmiu wykresów.

9. [P] Zaprezentuj prowadzącemu wyniki. Przygotuj się do omówienia wybranego przekodowania. (jak działa tj. jak przekształcane są piksele jasne, a jak ciemne).

Operacje arytmetyczne:

DODAWANIE

1. Utwórz nowy m-plik (**New->Script**) lub (**New Script**). Nazwij go i zapisz. Nie zapomnij o poleceniach `clearvars; close all; clc;`. Wczytaj dwa obrazy 'lena.bmp' i 'jet.bmp' i wyświetl je.

2. Dodaj obrazy **Lena** i **Jet,** wykorzystaj funkcję `imadd` (sposób jej użycia należy sprawdzić w pomocy Matlaba). Uzyskany wynik wyświetl.

3. Czy wynik sumowania jest satysfakcjonujący?. Co może niekorzystnie wpływać na rezultat operacji ? Funkcja `imadd` ma możliwość podania typu danych w jakim może być zapisany wynik. Spróbuj wykorzystać typ **uint16.** Uwaga do poprawnego wyświetlania potrzebna jest następująca modyfikacja: `imshow(sum,[]);`. Parametr [] oznacza, że dane z obrazu **sum** zostaną przed wyświetleniem przeskalowane do zakresu 0-255, przy czym jako 0 zostanie wzięte min(**sum**), a jako 255 max(**sum**). **Uwaga**: operacja ta jest użyteczna w przypadku gdy dane do wyświetlenia wykraczają poza zakres 0-255, w przeciwnym przypadku jej wykorzystanie może zniekształcić wyniki.

KOMBINACJA LINIOWA

4. Do wykonywania operacji kombinacji liniowej służy funkcja `imlincomb`. Zapoznaj się z dokumentacją tej funkcji i przetestuj kilka kombinacji liniowych obrazów **Lena** i **Jet.**

ODEJMOWANIE

5. Wykorzystując funkcję `imsubtract` odejmij obrazy **Lena** i **Jet.**

6. Czy wynik odejmowania jest satysfakcjonujący ? Co może niekorzystnie wpływać na rezultat operacji ? Rozwiązaniem problemu jest zmiata typu danych dla obrazów **Lena** i **Jet** z **uint8** na **int16**. Odpowiedź na pytanie dlaczego ta zmiana poprawia wynik odejmowania ? Przydatna składnia: `lena16 = int16(lena);` Podczas wyświetlania pamiętaj o przeskalowaniu ([]).

7. Często zamiast zwykłego odejmowania wykorzystuje się operację wartość bezwzględna z różnicy (pozwala to m. in. uniknąć pokazanych powyżej problemów). Wykorzystując funkcję `imabsdiff` wykonaj operację wartość bezwzględna z różnicy dla obrazów **Lena** i **Jet.**

MNOŻENIE

8. Mnożenie dwóch obrazów pozwala wykonać funkcja `immultiply`. Wykonaj mnożenie obrazów **Lena** i **Jet** - czy wynik takiej operacji zawiera jakąś istotną informację ? Dlaczego?

9. Mnożenie częściej wykorzystuje się jako
   - mnożenie przez stałą - co powoduje ogólne rozjaśnianie albo ściemnianie obrazu,
   - mnożenie przez maskę - czyli obraz binarny.

10. Przetestuj na wybranym obrazie mnożenie przez stałą. Następnie wczytaj maskę 'kolo.bmp'. Zamień wczytaną macierz na typ boolean (np. `maska = boolean(maska);`) Przemnóż wybrany obraz przez maskę.

NEGATYW

11. Często wykorzystywaną operacją jest negatyw (pokazany wcześniej przy okazji operacji LUT) - funkcja `imcomplement`. Przetestuj jej działanie.

12. [P] Zaprezentuj wyniki prowadzącemu.

Operacje logiczne:

Na poszczególnych punktach obrazu (najczęściej binarnego - czyli składającego się z dwóch kolorów: czarnego i białego) można wykonywać operacje logiczne: NOT, AND, OR, XOR itp.

1. Utwórz nowy m-plik (**New->Script**) lub (**New Script**). Nazwij go i zapisz. Nie zapomnij o poleceniach `clearvars; close all; clc;`. Wczytaj dwa obrazy 'kolo.bmp' i 'kwadrat.bmp'.

2. Zamień wczytane obrazy na typ boolean (np. `kolo = boolean(kolo);`). Wyświetl wczytane obrazy.

3. Na wczytanych obrazach wykonaj wybrane operacje logiczne. NOT (operator '~'), AND ('&'), OR ('|'), XOR (`xor`). Rezultaty wyświetl.

4. [P] Wyniki zaprezentuj prowadzącemu.